N74-10182

20029-H026-R0-00

TRW NOTE NO. 72-FMT-891

PROJECT APOLLO

TASK MSC/TRW A-527

AUTOMATED VERIFICATION SYSTEM USER'S GUIDE

12 JANUARY 1972

Prepared for
MISSION PLANNING AND ANALYSIS DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS
NAS 9-12330

20029-H026-R0-00

TRW NOTE NO. 72-FMT-891

PROJECT APOLLO

TASK MSC/TRW A-527

AUTOMATED VERIFICATION SYSTEM
USER'S GUIDE

12 JANUARY 1972

Prepared for
MISSION PLANNING AND ANALYSIS DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS
NAS 9-12330

Prepared by R. H. Hoffman
R. H. Hoffman
Simulation Development Section

Approved by K. W. Krause
K.W. Krause, Task Manager
Head, Simulation Development Section

Approved by E D Stuckle
E.D. Stuckle, Manager
Software Systems Department

Approved by D P Johnson
D.P. Johnson
Assistant Project Manager
Mission Trajectory Control Program

TRW
SYSTEMS

ACKNOWLEDGMENTS

The author wishes to acknowledge the contributions of R. W. Smith and R. E. Allenson, TRW, in the development and documentation of the Automated Verification System and of C. E. Allday, Program Development Branch, MSC, in the direction of this project and in the compilation of this User's Guide.

ABSTRACT

This program technical report is submitted to NASA/MSC by TRW Systems Group in accordance with MSC/TRW Task A-527 of the Mission Trajectory Control Program, Contract NAS 9-12330. The contents of this guide include a description of the operational requirements for all of the programs of the Automated Verification System (AVS). The AVS programs are:

- FORTRAN code analysis and instrumentation program (QAMOD)
- Test Effectiveness Evaluation Program (QAPROC)
- Transfer Control Variable Tracking Program (QATRAK)
- Program Anatomy Table Generator (TABGEN)
- Network Path Analysis Program (RAMBLE)

(NASA-CR-115374) AUTOMATED VERIFICATION SYSTEM USER'S GUIDE (TRW Systems Group) ~~78~~ p HC CSCL 09B

73

N74-10182

Unclas

G3/08 21787

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

CONTENTS (Continued)

TABLES

PRECEDING PAGE BLANK NOT FILMED

ILLUSTRATIONS

PRECEDING PAGE BLANK NOT FILMED

GLOSSARY

Terms	Definition
AVS	Automated Verification System
ATDG	Automated Test Data Generation Subsystem
Automated Tools	See "Programming Tools"
Branch Control Variable	A variable that appears in a "branch expression"
Branch Expression	The condition whose value determines a transfer in a FORTRAN branch statement
Element	A FORTRAN main program, subroutine or function
FASTRAND	UNIVAC 1108 magnetic drum storage units which provide large capacity random access mass storage
High Speed Drum	UNIVAC 1108 FH-432 or FH-1782 (Flying Head) high speed large-capacity magnetic drum storage units which provide modular auxiliary storage
Instrumentation	Processing of FORTRAN code by QAMOD which involves the inserting of calls to an execution monitoring subroutine at each transfer statement and each addressable statement to record the execution of the statements
Path	A route through a FORTRAN "element" which is determined by interpretation of a series of "segment relationships"
Path Network	The set of all "paths" through an "element"
Program Anatomy Table	The data generated by TABGEN which contains a description of the "segment relationships," "branch expressions" and input/output variables within an "element."
Programming Tools	FORTRAN programs which analyze the syntax and logic structure of other programs to provide automated assistance in verification, maintenance, and documentation
Pseudo Number	See "Pseudo Statement Number"

Terms	Definition
Pseudo Statement Number	A number assigned to each executable statement by QAMOD to facilitate referencing of the statement by other AVS programs; the numbers are reinitialized at one for each element and are sequentially assigned
Recording File	A data file which records the execution of each statement (or subroutine) during execution of an "instrumented" program
Segment of Code	The smallest set of consecutively executable statements within a subroutine to which control can be given during program execution; the first statement is addressable and the last is a transfer to a different segment
Segment Relationship	The relationship between two "segments of code" resulting from a possible transfer of control which may occur during program execution
Statement Execution Recording File	See "Recording File" above
TDEM	Test Data Effectiveness Measurement Subsystem
Tools	See "Programming Tools"
Transfer Variable	See "Branch Control Variable"
Transfer Expression	See "Branch Expression"
Trap	v. the insertion of "instrumentation" before and/or after a statement to monitor its execution n. the call to an execution monitoring subroutine which is inserted during "instrumentation" of FORTRAN code
Unit Module	The smallest testable block of code within a program; a unit module will be a subprogram or portion of subprogram

1. INTRODUCTION

The Automated Verification System (AVS) is designed to satisfy two objectives: to reduce the cost of FORTRAN software by eliminating or supplementing existing tedious manual verification functions with automated aids; and to produce, through the use of automated aids, means by which test effectiveness can be measured, thus providing measurable proof of software quality. Although these objectives can be realized, it is clear that automation can only aid the verification effort. The validity of test results can only be determined by the testor or user.

The AVS is presently composed of two subsystems: the Automated Test Data Generation (ATDG) Subsystem and the Test Data Effectiveness Measurement (TDEM) Subsystem. The objectives of the ATDG Subsystem are to generate an optimal set of test cases by analyzing the structure and variables of a subject FORTRAN program and measure the effectiveness of the test cases based on branches and paths exercised. Only a portion of this capability is operational, the utility routines required as a basis for this capability. Only this operational capability is described in this user's guide. As the remaining ATDG Subsystem is developed this document will be updated.

The TDEM Subsystem is completely operational and its usage is described in this document. It may be used to measure the frequency of execution of code in a subject FORTRAN program for a set of test cases, identify the unexercised code, and provide information to the user to modify his test data and effect execution of the unexercised code.

The Automated Verification System is being developed for the Program Development Branch of the Mission Planning and Analysis Division under MSC/TRW Task A-527. Several basic elements of the system were originally developed by the TRW Systems Product Assurance organization as part of the Product Assurance Confidence Evaluator (PACE) System. FLOW, one of the PACE programs, served as the basic element of the TDEM Subsystem.

The AVS can be used as an entity to verify subject programs or modules, the two subsystems can be applied individually, or specific elements of ATDG or TDEM can be used separately. The intended usage is consistent with the verification guidelines of Reference 1. That is, it is recommended that the user should begin with exhaustive testing at the unit module level.

Test planning and ultimately test data for each module may be generated with the ATDG Subsystem. The effectiveness of each test case may then be measured with the TDEM Subsystem and/or the ATDG Subsystem. Having satisfactorily completed all unit module testing, data cases for the complete program may be generated in a similar manner, treating each unit module as a "black box" and verifying only the interfaces. The effectiveness of the interface testing may then be evaluated with the TDEM Subsystem.

The TDEM Subsystem may also be used to measure the effectiveness of an existing file of test cases and provide information for the set of test cases at the unit module or program level.

Based upon comments received from MSC and TRW users concerning the TDEM User's Guide (Reference 2), this document has been organized to accommodate the novice as well as the experienced user. Due to the intended utility of the Automated Verification System, it is desirable that a user should be able to obtain the fundamental test data evaluation and/or logical path network definition with only a minimal familiarity with the many optional features of the AVS. However, it is also necessary to include explanations of the more complex features to fulfill the requirements of more experienced users. This has been accomplished by providing basic information and optional features in two separate sections of this document.

This guide consists of five sections as follows:

- Introduction - a description of the format of the document and an explanation of the objectives, evolution, and future of the AVS programs;
- System Description - an overview of the AVS structure and capabilities;
- System Limitations - a discussion of programs which can be processed by the AVS;
- Basic User Information - a concise summary of the minimum inputs required for AVS operation;
- Optional Features - a complete description of all of the available AVS options.

The following steps are recommended for effective use of this document:

- Read the introduction in Section 1 and the system description on Section 2 to determine if the AVS capabilities will fulfill all requirements.

- Decide which elements of the TDEM and ATDG subsystems should be selected.

- Read Section 3 to insure that the subject program is acceptable to the AVS, taking note of any violations.

- Set up the AVS execution deck by following the appropriate sample deck illustration in Section 4.

- If any violations of the AVS limitations were noted in Section 3, refer to Section 5 for a description of the optional input parameters. Assign new values as required.

- Refer to the applicable "optional features selection chart" in Section 5 to assure that all necessary options have been employed.

- Assemble and execute the deck(s) to obtain fundamental AVS displays.

- If the quantity or format of the output is unsatisfactory, refer to Section 5. Options are provided to reduce or expand the level of detail of both the analyses and the displays.

2. SYSTEM DESCRIPTION

The capabilities which are operational in the AVS are as follows:

- FORTRAN source code is statically analyzed and instrumented to generate a chronological recording of statements exercised during execution of the subject program.

- By analysis and extrapolation of the logical transfers in each unit module of source code, all logical paths within the unit module are determined and displayed. The minimum characteristic subset of these paths is then automatically selected and is designated as the optimum set of paths from which a unit test data base may be constructed.

- Test efficiency statistics are derived based upon the statements exercised during the execution of the subject program with its test data base.

- Statements not exercised by the test data are displayed with the conditions necessary to effect their execution. The computations of the variables required to satisfy these necessary conditions are then displayed.

The AVS consists of five FORTRAN programs, each of which requires data generated by the other AVS programs and/or generates data required by other programs. Interprogram communication is via external data files which may be assigned to either magnetic tape or FASTRAND drum. The syntax analysis program QAMOD is the basic element of the system and is the first program to be executed in both the ATDG and TDEM Subsystems. For the ATDG Subsystem, QAMOD generates data required by the TABGEN and RAMBLE Programs. For the TDEM Subsystem, QAMOD generates data required by the QAPROC and QATRAK Programs.

2.1 AUTOMATED TEST DATA GENERATION (ATDG) SUBSYSTEM

The ATDG Subsystem is comprised of the code analyzer, QAMOD, the Program Anatomy Table generator, TABGEN, and the Network Path Analysis Program, RAMBLE. Figure 2-1 illustrates the program and data file interface of the ATDG subsystem. The files referenced in the figure are defined in Section 4.

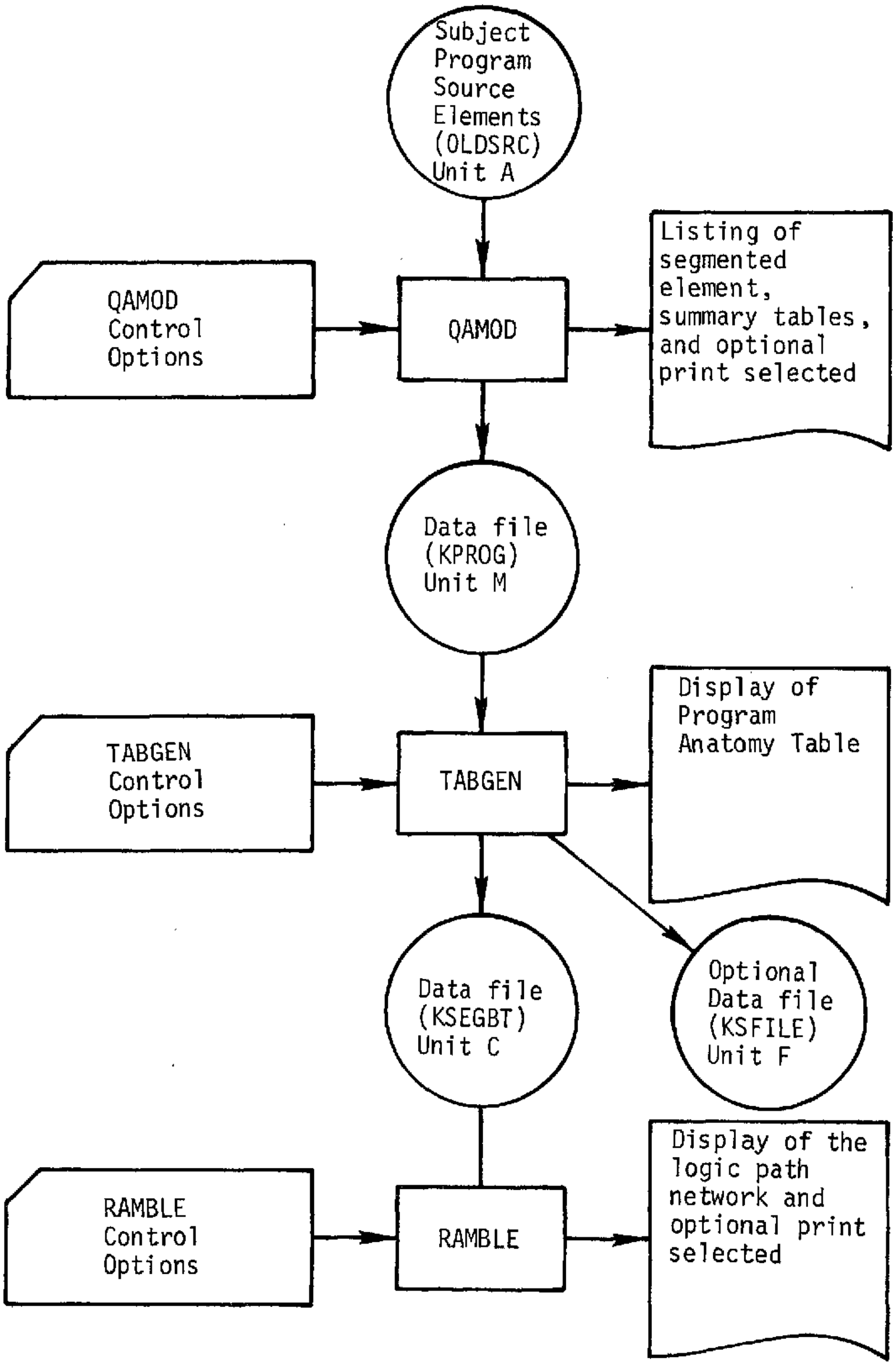

Figure 2-1. ATDG Subsystem Interface

2.1.1 QAMOD

The QAMOD Program sequentially analyzes each statement of a subject FORTRAN source element (i.e., subroutine, main program, or function), which may be input via cards or UNIVAC 1108 Program Complex File (PCF) tape. As each statement is read, the following analysis is accomplished:

- A syntax analysis is performed to determine the type of the statement (i.e., assign, transfer, conditional transfer, etc.).
- If, during execution of the subject code, control can be transferred to the current statement from a location other than the statement immediately preceding it, then the current statement is designated as the beginning of a segment of code.
- If, during execution of the subject code, control can be transferred to a location other than the statement immediately following the current statement, then the current statement is designated as the end of a segment of code.

After completion of this analysis for each statement, the statement is output with its type and segment number to the data file KPROG. When the entire element has been processed, information regarding the number of lines processed, the number of executable statements analyzed, and the DO loops within the element is also output to the data file KPROG.

2.1.2 TABGEN

The TABGEN Program performs syntax analysis which supplements the analyses accomplished by QAMOD and builds a Program Anatomy Table. The source statements with their designated type and segment number are read from the data file, KPROG; and, using these data together with variable usage and logical transfer information derived from the source statement, TABGEN generates a data file, KSFILE. KSFILE contains information required for definition of logical path structure and test data generation. TABGEN also has the capability of extracting the logical transfer data relative to each segment of code and generating a KSEGBT file for input to the RAMBLE program. The KSEGBT data consists of a table of all segment relationships (i.e., for each segment, the table lists all segments to which control can be transferred during execution of the subject program), segments which are entry points of the module, and segments which are terminal points of the module.

2.1.3 RAMBLE

RAMBLE stores the segment relationship data from the KSEGBT file and constructs all logical paths of the subject module in the form of segment strings. Two types of paths are generated: (1) those that begin at an entry point and end at an exit point of the module and (2) those that begin at an entry point and end at a segment which has previously appeared in the segment string (repeated segment). The latter type is referred to as a loop path.

Nominally RAMBLE will display all paths in the subject module. The display of loop paths in the network may be optionally suppressed.

RAMBLE has optional capability to perform characterization analysis. It first identifies the characteristic paths in the subject module. The characteristic paths have the following properties:

- Collectively, they are the minimum set of paths of the module which contain all the segment relationships of the paths which end in an exit segment.
- They are identified in order of importance; that is, the first characteristic path contains the most frequent and largest number of segment relationships, the second characteristic path has the same attributes for segment relationships not contained in the first, and so on until the last of the segment relationships are acknowledged.
- Each characteristic path is representative of a family of paths of the subject module based on commonality of segment relationships.

RAMBLE then groups all paths of the module into families according to their commonality with the characteristic path representing the family. Finally, RAMBLE performs a characterization of the loop paths within each family, identifying characteristic loops if they contain new segment relationships not present in the set of characteristic paths or previously identified characteristic loops. The algorithm for identifying the characteristic loops is similar to that for identifying characteristic paths. Therefore, the characteristic loops have similar properties (minimum set and hierarchy of importance).

Thus, a minimum set of paths and loops are identified to exercise all segment relationships in the subject module. The obvious application is to

utilize the hierarchy to generate the optimal set of test cases for exhaustive testing of the module.

2.1.4 Applications of ATDG

The intended utility of the ATDG Subsystem is to provide automated analysis of a unit module to determine the test cases which will most effectively test the logical paths of the module. By processing the module with QAMOD, TABGEN, and RAMBLE, the user is provided with a detailed description of the logic network and a display of the paths which must be executed to assure that all logical transfers are exercised. Thus aids in optimal test planning are provided by these utility tools. As the remaining programs and routines of the ATDG are developed more detail will be provided to the user to attain the goal of automated optimal test data generation.

2.2 TEST DATA EFFECTIVENESS MEASUREMENT (TDEM) SUBSYSTEM

The TDEM Subsystem is comprised of the code analysis and instrumentation program, QAMOD, the test data evaluator, QAPROC, and the program, QATRAK, which displays the computation or input of the transfer variables to be changed to effect execution of the unexercised statements. Figure 2-2 illustrates the program and data file interface of the TDEM subsystem. The files referenced in the figure are defined in Section 4.

2.2.1 QAMOD

The QAMOD program sequentially analyzes each statement of a FORTRAN source program, which may be input via cards or Univac 1108 Program Complex File (PCF) tape. As each statement is read, the following analysis is accomplished:

- The first executable statement of each element (i.e., subroutine or main program) is assigned a pseudo statement number of one. Each subsequent statement is assigned a sequential pseudo number and the statements are displayed with their pseudo numbers. Statements are later referenced from QAPROC and QATRAK by element name and pseudo number.

- The code is instrumented by the insertion of traps to monitor statement execution. The function of these traps is the generation of a statement execution recording file during execution of the instrumented program. The recording file registers the execution of each statement and the order of execution.

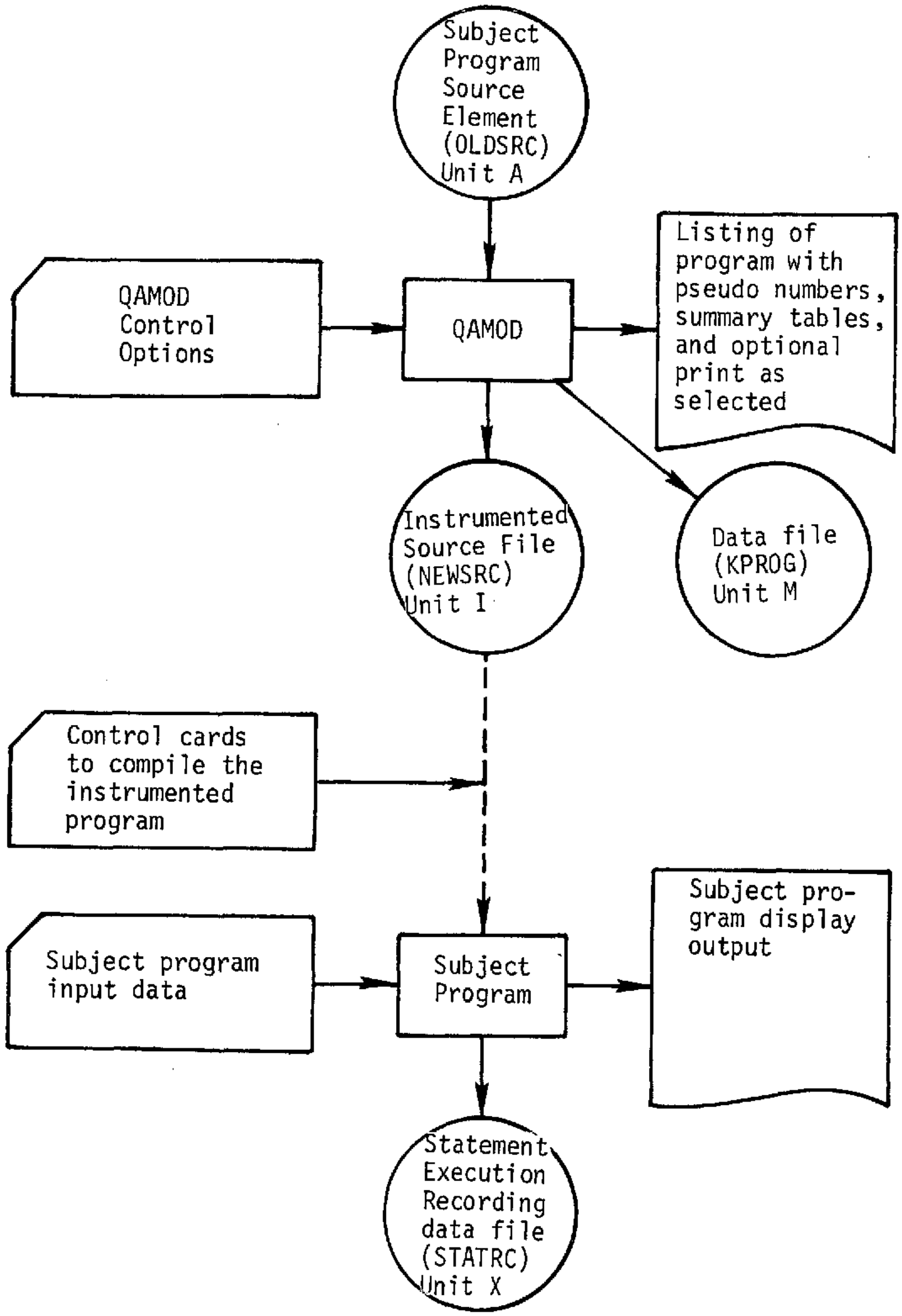

Figure 2-2. TDEM Subsystem Interface

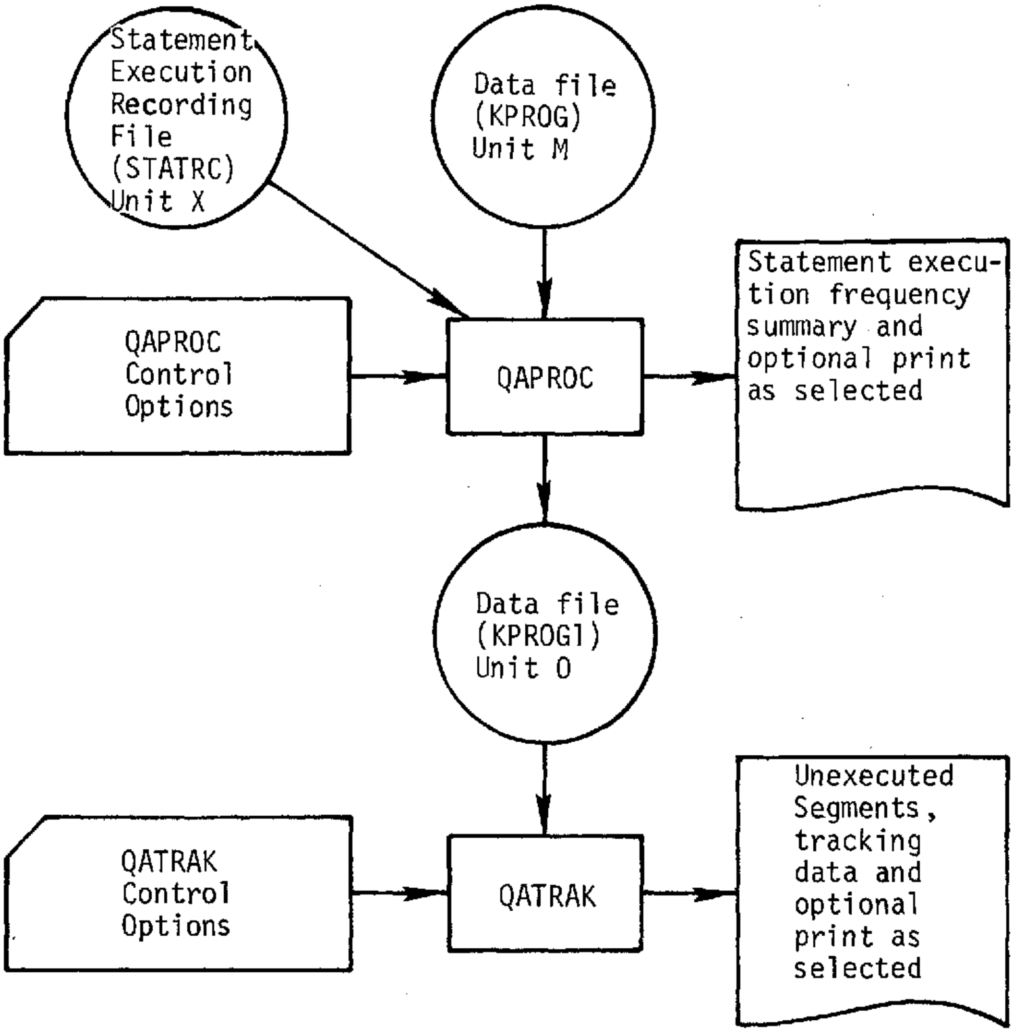

Figure 2-2. TDEM Subsystem Interface (Continued)

After completion of the analysis of each statement, the instrumented source program is output to a file, NEWSRC, for compilation and execution. Except for the generation of the recording file, this program executes exactly as the original subject program.

As the program is being processed, a data file KPROG is also generated. The KPROG file consists of information describing each statement and information relative to program size and structure.

2.2.2 QAPROC

The QAPROC Program accesses the statement execution recording file generated by execution of the instrumented subject program and produces an evaluation and summary of the test case executed. The recording file is sequentially accessed and the data are assimilated into an internal table, MAPTAB. At times designated by the input control options, a display is printed which includes:

- A map, delineated by subroutine, indicating the number of executions which have been recorded for each statement;
- Statistics indicating the percentage of the total executable statements which were executed;
- Statistics indicating the percentage of the total number of subroutines which were executed;
- A list of the names of subroutines which were not executed.

After processing the entire recording file, statement usage frequency information is added to the data from the input KPROG file and this revised information is output on a data file KPROG1.

Statistics from several recording files (i.e., several executions of the subject program) may be summed and a cumulative summary compiled. This accumulation is accomplished through utilization of the QAPROC optional features described in Section 5.

2.2.3 QATRAK

The QATRAK program accesses the KPROG1 data file generated by QAPROC and displays information indicating the variables whose values could be changed to provide more comprehensive verification of the subject program.

For each unexecuted block, or segment of code, the following information is displayed:

- A listing of the statements in the segment;
- An explanation of why the segment was not executed;
- All statements from which a transfer could be made to the segment;
- All computations of the branch control variables involved in the decision to transfer to the segment.

Through the optional track-back feature described in Section 5, the user may also select to track the computation and input of the branch control variables within all subroutines executed prior to the segment being analyzed.

The QATRAK displays provide aid to the user in the selection of new input data to effect execution of the unexercised segments, thereby generating a more comprehensive test data base.

2.2.4 Applications of TDEM

The TDEM Subsystem provides a meaningful measure of the effectiveness of test data and aids modification of the data to achieve more comprehensive verification of software. After applying TDEM to assure that 100 percent of the executable statements in the subject program have been exercised, the user will have generated a basic test data base. Although TDEM verifies that all statements have been executed, it does not assure that all of the functional requirements of the module have been satisisfied. This assurance can only be made by the user.

3. LIMITATIONS

3.1 ATDG

Since the intent of the ATDG Subsystem is the processing of a single element, there are no significant restrictions in the execution of QAMOD or TABGEN. The logic structure of the element is restricted, however, when applying RAMBLE to generate the network paths. The dimensions of the internal arrays in RAMBLE dictate the following maximums:

- 5000 paths
- 500 segments
- 500 segment relationships

If the element violates these maximums, the only solution is to divide the elements into several workable sized elements.

Another limitation of the operational RAMBLE utility routine is that <u>all</u> logical paths are identified. Capability of the routine is not yet available to discard paths containing segment relationships which are incompatible.

3.2 TDEM

The only general restriction to be considered when applying TDEM is the size of the subject program. Due to the dimensions of internal arrays in QAMOD, the subject program may not have more than:

- 7000 executable statements
- 500 DO loops
- 400 subroutines
- 500 entry points

If the program to be analyzed does not conform to the above size constraints, either of two solutions is possible:

- The program may be partitioned into portions which conform to all limitations. Each portion may then be processed separately as explained under "Element Specification Cards", Section 5.3.3.

- Instead of monitoring the execution of each statement, the monitoring may be performed at a less detailed level as explained under "LEVEL", Section 5.3.3.

After the subject program has been instrumented by QAMOD, the core storage necessary for its execution will increase by 30 to 80 percent, depending upon the amount of instrumentation required. This problem can usually be solved by overlaying the program, but if a satisfactory overlay cannot be accomplished then the solutions explained in the previous paragraph should be employed.

All other TDEM restrictions are minor and can be resolved by referring to the "Optional Features Selection" charts, Figure 5-1 and 5-2. The user should first read Section 4, however, and set up the nominal TDEM input check before proceeding to Section 5.

4. BASIC USER INFORMATION

The Automated Verification System (AVS) was designed and developed to be flexible enough to permit utilization of it on the widest possible range of FORTRAN programs. To attain this generality and still avoid overcomplication of the input required for the majority of users, the nominal and the optional AVS capabilities are described in separate sections of this document.

Section 4.1 presents a general description of the AVS programs' nominal input and output. For most applications, this general description with the deck setups illustrated will be all that is required. Section 3 explains the necessary criteria to determine if the subject program can be processed using the nominal values of the AVS input parameters. If the program does not conform to all of the limitations explained in Section 3, references to the explanations of the applicable program options are provided.

Having assembled the nominal deck by the appropriate illustration the user should then refer to the appropriate "optional features selection chart" in Section 5. These charts provide a step-by-step procedural checklist to assure that all necessary options have been set. Frequent reference to the chart, even by the experienced AVS users, will save both computer time and man-hours required for AVS applications.

4.1 INPUT/OUTPUT DESCRIPTION

The AVS programs described in this section are utilized by the TDEM and/or the ATDG Subsystems. For this reason, the title of each subsection indicates both the name of the program being described and the name(s) of the applicable subsystem(s).

4.1.1 QAMOD (TDEM and ATDG)

The QAMOD program is the basic element of the AVS, whether the application requires the TDEM or ATDG Subsystem. Since the QAMOD input and output are similar for either application, the information in this section pertains to both subsystems. The deck setup for the ATDG application is illustrated in Figure 4-1a and for the TDEM application in Figure 4-1b. Use of the QAMOD optional features for the TDEM application are illustrated in the optional features selection chart, Figure 5-1. The ATDG application is not illustrated, since only the parameter ISEG is required.

```
CC 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

 1  ▽  A S G   M = S A V E              QAMOD output file KPROG
 2  ▽  A S G   A = . . . . . .          Subject program source file OLDSRC
 3  ▽  A S G   I = W O R K              QAMOD output file NEWSRC
 4  ▽  A S G   N = . . . . . .          AVS programs file
 5      X Q T   C U R
 6      T R W   M , A , I , N            Rewind all files
 7      E R S                          }
 8      I N   N                        } Load AVS and release unit N
 9      T R I   N                      }
10  ▽  X Q T   Q A M O D                Execute QAMOD
11    $ M O D I N                      } NAMELIST input (Set ISEG to
12      I S E G = 1                    }   indicate ATDG application)
13    $ E N D
14 P R O G R 1                          Specify element to process
15 * B E G I N                          Begin QAMOD processing
```

Figure 4-1a. QAMOD Deck Setup (ATDG)

CC 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	
▽ ASG X=SAVE2	Statement execution recording file STATRC
▽ ASG M=SAVE	QAMOD output file KPROG
▽ ASG A=......	Subject program source file OLDSRC
▽ ASG I=SAVE1	QAMOD output file NEWSRC
▽ ASG N=......	AVS programs file
▽ XQT CUR	
TRW X,M,A,I,N	Rewind all files
ERS	Load AVS and release unit N
IN N	
TRI N	
▽ XQT QAMOD	Execute QAMOD
$MODIN	NAMELIST input (accept nominal values for all parameters)
$END	
*ALL	Process all elements
*BEGIN	Begin QAMOD processing
▽ XQT CUR	
IN I	Load NEWSRC file
▽ FOR,* SUBA,SUBA	Compile all of the elements which have been processed
▽ FOR,* SUBB,SUBB	
.	
.	
.	
▽ XQT CUR	

Figure 4-1b. QAMOD Deck Setup (TDEM)

CC	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	
24	TRW I	
25	OUT I	Save recompiled elements on NEWSRC file
26	TEF I	
27	TRW A,I	Erase PCF area and reload old source and new source to assure inclusion of elements not on NEWSRC file (MAP, ASM, etc.)
28	ERS	
29	IN A	
30	IN I	
31	TOC	Display table of contents
32	TRW I	
33	OUT I	Save complete new source on NEWSRC file
34	TEF I	
35	TRI A,I	Release units A and I
36 ∇	XQT MAIN	Execute new source
37	.	
38	.	Data required for execution of the subject program
39	.	
40 ∇	XQT CUR	
41	TRI X,M	Release all units

Figure 4-1b. QAMOD Deck Setup (TDEM) (Continued)

All controls of output format, input/output files, and the type and detail of analysis performed are specified via the NAMELIST $MODIN. For the nominal case, only the NAMELIST specification ($MODIN and $END) is necessary, since the nominal values assigned in QAMOD are sufficient. The optional features which may be selected are specified between the $MODIN and $END cards as explained in Section 5.3.

Following the $END card, a card with *ALL is input for TDEM as illustrated in Figure 4-1b, or an element name card for ATDG as illustrated in Figure 4-1a.

To indicate the QAMOD processing and to indicate the end of control specifications, a card is input with *BEGIN. In the majority of cases, the *BEGIN card is the last of the QAMOD input. However, if the subject source program is being input via cards instead of tape, these source cards are input after the *BEGIN card as explained in Section 5.3.

The nominal display output of QAMOD consists of an input definitions table, a listing of the subject program with pseudo numbers assigned to the executable statements, and a summary table describing the structure of the subject program (number of statements, number of entry points, etc.). Optionally, the user may select the display of the output data file, KPROG, which will suppress the nominal listing of the subject code.

The two data files output from QAMOD are the KPROG and the NEWSRC files. The KPROG file, which is utilized by both the TDEM and ATDG Subsystems, contains information relative to the overall structure of the subject program and to each executable statement within the program. The NEWSRC file is only applicable during operation of TDEM. NEWSRC is a compilable source file consisting of the original subject source program with the instrumentation added to provide the statement execution recording file required for test data effectiveness measurement.

Due to the large amount of display output available from QAMOD, only a representative portion of the nominal displays is illustrated in Table 4-1. The notation "⋮" indicates a portion of the display has been omitted.

If the optional feature to display the contents of the KPROG file is selected, the subject program listing in the nominal display is suppressed and the KPROG file is listed as shown in Table 4-2.

```
         ***** QAMOD INPUT DESCRIPTION *****
  NAME   VALUE      DEFINITION

 KRE        5    SYSTEM CARD READ FILE
 KPR        6    SYSTEM PRINT FILE
 KPN       -3    SYSTEM PUNCH FILE
 OLDSRC     1    INPUT SUBJECT PROGRAM SOURCE TAPE
 NEWSRC    11    OUTPUT INSTRUMENTED SOURCE TAPE
 NFILES     1    NUMBER OF FILES ON OLDSRC TAPE
 KPROG     15    PROGRAM STRUCTURE DATA OUTPUT FILE
 NOPRNT     0    DISPLAY NOMINAL PRINT
 KLIST      0    DO NOT DISPLAY KPROG DATA
 ISEG       1       SEGMENTATION IS BEING PERFORMED
 TRPEND     0    DO NOT INSTRUMENT FORTRAN 'END' STATEMENTS
 IFA        0    INSTRUMENT ARITHMETIC 'IF' AS 1 STATEMENT
 NINOPT     1    INPUT SOURCE ELEMENTS FROM TAPE
 NOTOPT     1    OUTPUT SOURCE ELEMENTS TO TAPE
 LEVEL      1    INSTRUMENT SUBJECT PROGRAM AT LEVEL 1
 LIMSEG     0    NO INTERMEDIATE INSTRUMENTATION LEVEL SELECTED
  SPECIFIED ELEMENTS ARE -
  BDSEL1
         ***********************************

ELEMENT BDSEL1                                                     0
C                                                                  0
      SUBROUTINE BDSEL1                                            0
C                                                                  0
      DIMENSION IBODYU(10)                                         0
            .                                                      .
            .                                                      .
            .                                                      .
      LEARTH = .FALSE.                                             5
      IF (IPLAN.NE.1) GO TO 30                                     5,    6
      IREG = 1                                                     7
      IREF = 1                                                     7
      IXLMAT = 1                                                   7
      IXREF = 1                                                    7
      GO TO 100                                                    7
   30 IF (IPLAN.EQ.4) LEARTH = .TRUE.                              8,    9
      NSAT = IPLCON(2,IPLAN)                                      10
      IF (ISAT) 40,40,60                                          10
   40 IREG = 3                                                    11
      IXLMAT = IPLAN                                              11
      IXREF = IPLAN                                               11
      IF (LEARTH) GO TO 50                                        11,   12
      IF (NSAT) 42,47,42                                          13
            .                                                      .
            .                                                      .
            .                                                      .
  600 CONTINUE                                                    76
      IF (ISUBPR(1).EQ.1) WRITE(6,710) IREFPL,IBODYA,IBODYC       76,   77
      RETURN                                                      78
      END                                                          0
            .
            .
            .
```

TABLE 4-1a. QAMOD Nominal Display (ATDG)

```
            ***** QAMOD INPUT DESCRIPTION *****
       NAME  VALUE    DEFINITION

      KRE       5   SYSTEM CARD READ FILE
      KPR       6   SYSTEM PRINT FILE
      KPN      -3   SYSTEM PUNCH FILE
      OLDSRC    1   INPUT SUBJECT PROGRAM SOURCE TAPE
      NEWSRC   11   OUTPUT INSTRUMENTED SOURCE TAPE
      NFILES    1   NUMBER OF FILES ON OLDSRC TAPE
      KPROG    15   PROGRAM STRUCTURE DATA OUTPUT FILE
      NOPRNT    0   DISPLAY NOMINAL PRINT
      KLIST     0   DO NOT DISPLAY KPROG DATA
      ISEG      0   NO SEGMENTATION IS BEING PERFORMED
      TRPEND    0   DO NOT INSTRUMENT FORTRAN 'END' STATEMENTS
      IFA       1   INSTRUMENT ARITHMETIC 'IF' AS 4 STATEMENTS
      NINOPT    1   INPUT SOURCE ELEMENTS FROM TAPE
      NOTOPT    1   OUTPUT SOURCE ELEMENTS TO TAPE
      LEVEL     1   INSTRUMENT SUBJECT PROGRAM AT LEVEL 1
      LIMSEG    0   NO INTERMEDIATE INSTRUMENTATION LEVEL SELECTED
       SPECIFIED ELEMENTS ARE -
       *ALL
              **********************************

ELEMENT MAIN                                                         0
      DIMENSION  STATE(8) , ITITLE(14) , X(3),Y(3),Z(3)              0
      COMMON /MODS/ MOD2,MOD3,MOD5,MOD7                              0
      DATA  STATE / 8*0. / , I1 / 8 / , I2 / 9 /                     0
      INPUT = 0                                                      1
      REWIND I1                                                      2
      READ (5,100)  ITITLE                                           3
  100 FORMAT ( 14A6 )                                                0
  110 READ(5,500) MODEL, (STATE(I),I=1,8)                            4
            .                                                        .
            .                                                        .
            .                                                        .
      CALL XYZ(X,Y,Z,MODEL)                                         27
      IF ( INPUT.LT.6 )   GO TO 110                                 28,     29
      STOP                                                          30
      END                                                            0

ELEMENT XYZ                                                          0
      SUBROUTINE XYZ(X,Y,Z,MODEL)                                    0
      DIMENSION  STATE(8),X(3),Y(3) , Z(3), XDOT(3),YDOT(3),ZDOT(3)  0
      COMMON /MODS/ MOD2,MOD3,MOD5,MOD7                              0
      DATA  IFLAG/0/                                                 0
C     IFLAG = 0 AT FIRST CALL                                        0
      IF (IFLAG.NE.0)  GO TO 200                                     1,      2
      IFLAG = 1                                                      3
      DO 100 I=1,3                                                   4
      XDOT(I) = 0.0                                                  5
            .                                                        .
            .                                                        .
            .                                                        .
      GO TO 2050                                                   238
 1000 CONTINUE                                                     239
 1100 CONTINUE                                                     240
      RETURN                                                       241
      END                                                            0

            ***** QAMOD SUMMARY PRINT *****
     THE 25 ELEMENTS ANALYZED CONTAINED THE FOLLOWING -
      2210 STATEMENTS, 1637 WHICH WERE TRAPPED AT LEVEL 1
        31 ENTRY POINTS
         3 STOP STATEMENTS
         0 END STATEMENTS
      1810 KPROG RECORDS
            *********************************
```

TABLE 4-1b. QAMOD Nominal Display (TDEM)

Preceding page blank

```
        *** PROGRAM CARD IMAGE (KPROG TAPE) LISTING ***

STATEMENT********************          / ELEMENT MAIN   /      PSEUDO NUMBERS  1,   2,   3,   4-FREQ.  1,   2,   3,   4-TYPE

ELEMENT MAIN
      DIMENSION  STATE(8) , ITITLE(14) , X(3),Y(3),Z(3)                         .                   .              .
      COMMON /MODS/ MOD2,MOD3,MOD5,MOD7                                         .                   .              .
      DATA  STATE / 8*0. / , I1 / 8 / , I2 / 9 /                                .                   .              5
      INPUT = 0                                                                 .                   .              6
      REWIND I1                                                                 1                   0              19
      READ (5,100)  ITITLE                                                      2                   0              0
  100 FORMAT ( 14A6 )                                                           3                   0              17
  110 READ(5,500) MODEL, (STATE(I),I=1,8)                                       .                   .              .
         .                                                                      4                   0              17
         .
         .
      CALL XYZ(X,Y,Z,MODEL)                                                     27                  0              3
      IF ( INPUT.LT.6 )   GO TO 110                                             28,  29             0,   OT,   OF   13
      STOP                                                                      30                  0              4
      END                                                                       .                   .              .
  .
  .
  .

     ***** PROGRAM STRUCTURE TABLES *****
       7         1
  100300       10016
  101000       20239
  .
  .
  .
      25          3          0        31       1810        1
MAIN        3000030          1
XYZ        24100271         26
 .             .             .
 .             .             .
INTG        1701637       1797
     ****************************************
```

TABLE 4-2. QAMOD Optional Display

4.1.2 TABGEN (ATDG)

TABGEN is the second program of the Automated Test Data Generation Subsystem. The nominal deck setup for TABGEN is illustrated in Figure 4-2 and optional features are explained in Section 5.1.

The TABGEN Program requires one input data file, KPROG, which is generated by QAMOD. From the KPROG information, TABGEN generates the Program Anatomy Table required by other programs in the ATDG subsystem. Although the primary purpose is to provide input to other elements of the ATDG Subsystem, the user may obtain information about the internal structure of his module from the Program Anatomy Table.

All controls of output format and input/output files are specified via the NAMELIST $TABIN. For the nominal case, only the NAMELIST specification ($TABIN and $END) is necessary, since the nominal values assigned in TABGEN are sufficient. The optional features which may be selected are specified between the $TABIN and $END cards as explained in Section 5.1.

Following the $END card, the element to be processed is specified as illustrated in Figure 4-2. Following the element specification, a card is input with *BEGIN to initiate the TABGEN processing and to indicate the end of control specifications.

Two data files may be output from TABGEN. The first file, KSEGBT, is the nominal output file containing the transfer data required by the ATDG program RAMBLE. The second, KSFILE, contains the Program Anatomy Table and may be output using the optional features explained in Section 5.1.

The nominal display output from TABGEN is shown in Table 4-3.

If the optional feature is selected to generate the KSFILE tape, the Program Anatomy Table is displayed. This table actually is comprised of six internal tables from the TABGEN Program. These tables are:

1. SEGTAB - each row contains information describing a particular segment including a segment identifier, segment number, its relative location within the element, its FORTRAN label, the segment type and branch expression type, and a pair of pointers indicating the location of pertinent information in the other five tables.

Preceding page blank

CC	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	
1	▽ ASG M=......	QAMOD output file KPROG
2	▽ ASG N=......	AVS programs file
3	▽ ASG C=SAVE	TABGEN output/RAMBLE input file KSEGBT
4	▽ XQT CUR	
5	TRW M,N,C	Rewind all files
6	ERS	Load AVS and release unit N
7	IN N	
8	TRI N	
9	▽ XQT TABGEN	Execute TABGEN
10	$TABIN	NAMELIST input (accept nominal values for all parameters)
11	$END	
12	MAIN	Specify element to process
13	*BEGIN	Begin TABGEN processing
14	CASEONE	Input title for the case
15	▽ XQT RAMBLE	Execute RAMBLE
16		Optional features specification card (nominally blank)
17	▽ XQT CUR	
18	TRI M,C	Release all units

Figure 4-2. TABGEN/RAMBLE Deck Setup (ATDG)

```
 ***** TABGEN INPUT DESCRIPTION *****
 NAME  VALUE    DEFINITION

KRE      5   SYSTEM CARD READ FILE
KPR      6   SYSTEM PRINT FILE
KSEGBT   3   FIRST OUTPUT DATA FILE (O=DO NOT OUTPUT)
KSFILE   8   SECOND OUTPUT DATA FILE(O=DO NOT OUTPUT)
 ***********************************


FAMBLE TABLES (KSEGBT) LISTING


BCSEL1  SEGMENTS BRANCHED TO TABLES (SEGBT) FOR RAMBLE INPUT
FRCM       1          TO          2
FRCM       2          TO          4
FRCM       2          TO          3
FRCM       3          TO          4
FRCM       4          TO          2
FRCM       4          TO          5
FRCM       5          TO          6
FRCM       5          TO          7
FRCM       6          TO          8
FRCM       7          TO         21
FRCM       8          TO          9
FRCM       8          TO         10
FRCM       9          TO         10
FRCM      10          TO         11
FRCM      10          TO         17
FRCM      11          TO         12
FRCM      11          TO         13
FRCM      12          TO         16
FRCM      13          TO         14
FRCM      13          TO         15
 .
 .
 .
FRCM      74          TO         76
FRCM      75          TO         76
FRCM      76          TO         77
FRCM      76          TO         78
FRCM      77          TO         78
STAR       1
STAR      22
END       78
SET
ECF
```

TABLE 4-3. TABGEN Nominal Display

SEGMENT TABLES FILE (KSFILE) LISTING

PROGRAM	DATE	TITLE
TABGEN	111171	FIRST TABGEN TEST ON BDSEL1

ELEMENT NAME	SEGTAB LENGTH	SEGBT LENGTH	SEGBF LENGTH	BETAB LENGTH	IVTABL LENGTH	OVTABL LENGTH
BDSEL1	78	115	115	241	118	99

SEGTAB TABLES

SEGID	SEGNUM	LINE-1	SEGLBL	LINE-N	SEGBTP	SBXTYP	SBRXP	SEGBFP	SIVP	SOVP	SEGTYP
BDSEL1	1	1	23	0	23	0001000001	7	0000000000	0000000000	0000000000	0001000001	1
BDSEL1	2	2	24	0	25	0002000004	4	0001000013	0001000002	0001000001	0002000002	6
BDSEL1	3	3	26	10	27	0005000005	7	0000000000	0003000003	0002000004	0003000004	5
BDSEL1	4	4	28	20	28	0006000007	10	0014000017	0004000006	0000000000	0000000000	5
:												

SEGBT 115 , 1

	1
1	2
2	4
3	4
4	3
:	

SEGBF 115 , 1

	1
1	1
2	4
3	2
4	2
:	

BETAB TABLE

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
I	P	L	C	O	N	(	1	,	I	)	I	,	1	0	I	P	L	A	N	.	N	E	.	1	I	P	L	A	N	.	E	Q	.	4	I	S	A	T	L
E	A	R	T	H	N	S	A	T	L	E	A	R	T	H	I	S	L	O	P	)	(	1	)	.	E	Q	,	1	I	,	1	0	I	,	2	1	I	,	1
:																																							

IVTABL TABLE

IPLCON	J	I	J	J	IPLAN	IPLCON	IPLAN	ISAT	IPLAN
IPLAN	LEARTH	NSAT	IPLAN	IPLCON	IPLAN	ISAT	LEARTH	ISAT	CONTIN
:									

OVTABL TABLE

L	I	IBODYU	J	NBODYL	LEARTH	IREG	IREF	IXLMAT	IXREF
IPLAN	NSAT	IREG	IXLMAT	IXREF	IREF	IREF	IREF	IREG	IXREF
:									

TABLE 4-4. TABGEN Optional Display

2. SEGBT - each segment to which a transfer can be made is included; the pointer SEGBTP designates the location of the "branch to" possibilities for a particular segment.

3. SEGBF - each segment from which a transfer can be made are included; the pointer SEGBFP designates the location of the "branch from" possibilities for a particular segment.

4. BETAB - each segment which ends with some type of branch (or transfer) has its branch expression included; the location of the expression for a particular segment is designated by the pointer SBRXP.

5. IVTABL - the input variables for a particular segment are included as designated by pointer SIVP.

6. OVTABL - the output variables for a particular segment are included as designated by the pointer SOVP.

Since these tables are primarily used internally in the ATDG Subsystem, they are not explained in detail in this document but are described in the AVS Programmer's Guide. The optional printed output from TABGEN is illustrated in Table 4-4.

4.1.3 RAMBLE (ATDG)

RAMBLE is the third program currently operational in the Automated Test Data Generation Subsystem. The nominal RAMBLE deck set-up is depicted in Figure 4-2 and options are explained in Section 5.2.

The RAMBLE program requires one input data file, KSEGBT, which is output from TABGEN. From the KSEGBT information, the path network displays are generated by RAMBLE.

The only input data card required by RAMBLE is the optional features specification card explained in Section 5.2. If nominal output is sufficient, only a blank card is input.

Due to the large amount of display output available from RAMBLE, only a representative portion of the displays is illustrated in Table 4-5.

If the OPTION field on the optional features specification card is input as 1, then the paths ending in other than a terminal segment will not be displayed (i.e., loop paths will be deleted from the output print). If

Preceding page blank

```
BDSEL1 SEGMENTS BRANCHED TO TABLES (SEGBT) FOR RAMBLE INPUT

         DATA  INPUT
FROM        1     TO          2
FROM        2     TO          4
FROM        2     TO          3
FROM        4     TO          2
  .         .     .           .
  .         .     .           .
  .         .     .           .
FROM       76     TO         77
FROM       76     TO         78
FROM       77     TO         78

TYPE=    1 OPTION=    0

INPUT OF RELATIONSHIPS IS BY TAPE

THE BEGINNING NODES ARE
           1
          22

THE ENDING NODES ARE
          78
     .
     .
     .
      LISTING OF ALL PATHS APPEARS HERE IN NOMINAL PRINT
     .
     .
     .
TOTAL NUMBER OF PATHS ENDING AT A SPECIFIED END SEGMENT= 1710
TOTAL NUMBER OF PATHS ENDING AT A REPEATED END SEGMENT =  857
TOTAL NUMBER OF PATHS DEFINED FOR THIS NETWORK         = 2567
```

TABLE 4-5. RAMBLE Nominal Display

```
THE CHARACTERISTIC PATHS OBTAINED FOR THE DESCRIBED NETWORK ARE

*    1*
       1 -     2 -     3 -     4 -     5 -     6 -     8 -     9 -    10 -    11 -    13
-     14 -    21 -    22 -    23 -    24 -    25 -    26 -    27 -    28 -    29 -    30
-     31 -    32 -    63 -    64 -    65 -    67 -    68 -    70 -    71 -    73 -    74
-     76 -    77 -    78

*    2*
       1 -     2 -     4 -     5 -     6 -     8 -    10 -    17 -    18 -    20 -    21
-     23 -    24 -    25 -    26 -    27 -    28 -    29 -    30 -    31 -    32 -    36
-     37 -    38 -    40 -    41 -    42 -    44 -    45 -    46 -    47 -    76 -    78

*    3*
       1 -     2 -     3 -     4 -     5 -     6 -     8 -     9 -    10 -    11 -    12
-     16 -    21 -    22 -    23 -    24 -    25 -    26 -    27 -    28 -    29 -    30
-     31 -    32 -    50 -    51 -    52 -    54 -    55 -    56 -    58 -    59 -    60
-     61 -    76 -    77 -    78

*    4*
       1 -     2 -     3 -     4 -     5 -     6 -     8 -     9 -    10 -    11 -    13
-     18 -    21 -    22 -    23 -    24 -    25 -    26 -    27 -    28 -    29 -    30
-     31 -    32 -    36 -    37 -    39 -    40 -    41 -    42 -    43 -    48 -    76
-     77 -    78

*    5*
       1 -     2 -     3 -     4 -     5 -     6 -     8 -     9 -    10 -    17 -    19
-     21 -    22 -    23 -    24 -    25 -    26 -    27 -    28 -    29 -    30 -    31
-     32 -    50 -    51 -    53 -    54 -    55 -    56 -    57 -    62 -    76 -    77
-     78

*    6*
       1 -     2 -     3 -     4 -     5 -     7 -    21 -    22 -    23 -    24 -    25
-     26 -    27 -    28 -    29 -    30 -    31 -    32 -    63 -    64 -    66 -    67
-     68 -    69 -    75 -    76 -    77 -    78

*    7*
       1 -     2 -     3 -     4 -     5 -     6 -     8 -     9 -    10 -    11 -    12
-     16 -    21 -    22 -    23 -    24 -    25 -    26 -    27 -    28 -    29 -    30
-     31 -    32 -    63 -    64 -    65 -    67 -    68 -    70 -    72 -    73 -    74
-     76 -    77 -    78

*    8*
       1 -     2 -     3 -     4 -     5 -     6 -     8 -     9 -    10 -    11 -    12
-     16 -    21 -    22 -    23 -    24 -    25 -    26 -    27 -    28 -    29 -    30
-     31 -    32 -    36 -    37 -    38 -    40 -    41 -    49 -    76 -    77 -    78

*    9*
       1 -     2 -     3 -     4 -     5 -     6 -     8 -     9 -    10 -    11 -    12
-     16 -    21 -    22 -    23 -    24 -    25 -    26 -    27 -    28 -    29 -    30
-     31 -    32 -    33 -    34 -    35 -    76 -    77 -    78

*   10*
       1 -     2 -     3 -     4 -     5 -     6 -     8 -     9 -    10 -    11 -    12
-     16 -    21 -    22 -    23 -    24 -    25 -    26 -    27 -    28 -    29 -    30
-     31 -    32 -    50 -    51 -    52 -    54 -    55 -    76 -    77 -    78
```

TABLE 4-6. RAMBLE Optional Display

```
*****CHARACTERISTIC PATH*****

*      1*
          1 -        2 -        3 -        4 -        5 -        6 -        8 -        9 -       10 -       11 -       13
*        14 -       21 -       22 -       23 -       24 -       25 -       26 -       27 -       28 -       29 -       30
*        31 -       32 -       63 -       64 -       65 -       67 -       68 -       70 -       71 -       73 -       74
*        76 -       77 -       78


*******************************************************************************************************************

             *****CHARACTERISTIC LOOP*****

*      1*
         70 -       71 -       73 -       70

*      2*
         64 -       65 -       67 -       64

*      3*
         30 -       31 -       30

*      4*
         28 -       29 -       28

*      5*
         26 -       27 -       26

*      6*
         24 -       25 -       24

*      7*
         51 -       53 -       54 -       51

*      8*
         33 -       34 -       33

*      9*
          2 -        3 -        4 -        2


*******************************************************************************************************************

* 10001*
          1 -        2 -        4 -        5 -        6 -        8 -        9 -       10 -       11 -       13 -       14
*        21 -       22 -       23 -       24 -       25 -       24

* 10002*
          1 -        2 -        4 -        5 -        6 -        8 -        9 -       10 -       11 -       13 -       14
*        21 -       22 -       23 -       24 -       25 -       26 -       27 -       26

* 10003*
```

TABLE 4-6. RAMBLE Optional Display (Continued)

CHARACTERISTIC PATH ANALYSIS SUMMARY

CHARACTERISTIC PATH	CHARACTERISTIC LOOPS FOR GROUP	PATHS GROUPED WITH CHARACTERISTIC PATH: ENDING AT A SPECIFIED END SEGMENT	ENDING AT A REPEATED END SEGMENT	TOTAL NUMBER
1	9	267	194	461
2	3	209	133	342
3	0	228	154	382
4	0	236	115	351
5	0	220	82	302
6	0	58	28	86
7	0	211	101	312
8	0	108	37	145
9	0	84	13	97
10	0	89	0	89

TABLE 4-6. RAMBLE Optional Display (Continued)

the TYPE field is input as 1, the characteristic paths will be printed with all of their related "family" paths as well as their characteristic loops as shown in Table 4-6.

4.1.4 QAPROC (TDEM)

QAPROC is the second program of the Test Data Effectiveness Measurement Subsystem. The nominal QAPROC deck setup is illustrated in Figure 4-3 and the optional features are explained in Section 5.4 and Figure 5-2.

After the instrumentation of the subject program is accomplished by QAMOD, the instrumented version (NEWSRC) is then executed with the desired test data, and a statement execution recording file, STATRC, is generated. The TDEM program, QAPROC, then accesses STATRC and calculates statistics relative to test data effectiveness measurement. Before accessing the recording file, the KPROG file program structure information is stored in QAPROC internal arrays.

All controls of output format, input/output files, and the types of statistics derived are specified via the NAMELIST $FLOWIN. For the nominal case, only the NAMELIST specification ($FLOWIN and $END) is necessary, since the nominal values assigned in QAPROC are sufficient. The optional features which may be selected are specified between the $FLOWIN and $END cards as explained in Section 5.4. The $END card is the final card in the QAPROC input deck.

The nominal display output (Table 4-7) includes:

- A description of the QAPROC input parameters as they have been specified by the user;
- A map table, delineated by subroutines, indicating the number of times that each statement was exercised in the execution of the test data;
- A summary table, which displays the statistics relative to percentage of the program executed.

Optionally, the contents of the output KPROG1 tape may also be displayed (Table 4-8a). A second option is the display of the logic trace of the statements exercised during execution of the instrumented subject program (Table 4-8b).

There are two data files, KPROG1 and SUBTRK, which may be output from QAPROC, but only the former is required by the majority of users. The

CC	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	
1	∇ ASG X = ...	Statement execution recording file STATRC
2	∇ ASG M = ...	QAPROC input file KPROG
3	∇ ASG A = ...	Subject program source file OLDSRC
4	∇ ASG N = ...	AVS programs file
5	∇ ASG O = SAVE	QAPROC output/QATRAK input file KPROG1
6	∇ ASG T,W	FASTRAND files for QATRAK
7	∇ XQT CUR	
8	TRW X,M,A,N,O,T,W	Rewind all files
9	ERS	
10	IN N	Load AVS and release unit N
11	TRI N	
12	∇ XQT QAPROC	Execute QAPROC
13	$FLOWIN	NAMELIST input (accept nominal values for all parameters)
14	$END	
15	IN A	Load subject program relocatable elements
16	TRW A	Rewind source file
17	XQT QATRAK	Execute QATRAK
18	$TRAKIN	NAMELIST input (accept nominal values)
19	$END	
20	*ALL	Process all elements
21	*BEGIN	Begin QATRAK processing
22	∇ XQT CUR	
23	TRI X,M,A, O,T,W	Release all units

Figure 4-3. QAPROC/QATRAK Deck Setup (TDEM)

```
        ***** QAPROC INPUT DESCRIPTION *****
 NAME   VALUE    DEFINITION

STATRC  27   STATEMENT EXECUTION RECORDING FILE
MAPSAV   0   OUTPUT MAP SUMMARY FILE
MAPIN    0   INPUT MAP SUMMARY FILE
BEGNTR   0   NUMBER PAIR AT WHICH TRACE BEGINS(0=NO TRACE)
ENDTR    0   NUMBER PAIR AT WHICH TRACE ENDS  (0=NO TRACE)
IMAP     0   MAP DISPLAY WILL BE OUTPUT
MAXPAR 100000 NUMBER PAIR AT WHICH TO DISPLAY FIRST MAP
INCPAR 200000 NUMBER OF PAIRS TO PROCESS BETWEEN MAPS
MAPMAX  10   MAXIMUM NUMBER OF MAP DISPLAYS REQUESTED
NOTAB    0   MAP SUMMARIES WILL BE CUMULATIVE
KLIST    0   DO NOT LIST KPROG1 FILE
NCASES   1   NUMBER OF CASES BEING PROCESSED
KTRACK   0   SUBTRK FILE WILL NOT BE OUTPUT
        ************************************

**QAPROC MAP PRINT**

--------------------
    ELEMENT MAIN
 PSEUDO NOS. FREQ. PSEUDO NOS. FREQ  PSEUDO NOS. FREQ  PSEUDO NOS. FREQ
    1 TO   3=   1      4 TO  29=   6     30 TO  31=   1

--------------------
    ELEMENT XYZ
 PSEUDO NOS. FREQ  PSEUDO NOS. FREQ  PSEUDO NOS. FREQ  PSEUDO NOS. FREQ
    1 TO   1=   6      2 TO   2=   5      3 TO   7=   1      8 TO   8=   6
       .
       .
       .

--------------------
**QAPROC USAGE SUMMARY AFTER    122462 NUMBER PAIRS(ENTRY/EXIT SEGMENTS)

  THE TEST DATA EXERCISED  1514 OF  1637 EXECUTABLE STATEMENTS.
  THE TEST EFFECTIVENESS RATIO AT THE STATEMENT LEVEL IS    .92

  THE PROGRAM CONTAINS  3 TERMINATION POINTS, ONLY ONE OF WHICH WAS
  EXECUTED. THE CORRECTED TEST EFFECTIVENESS RATIO IS   .93

  THE TEST DATA EXERCISED  26 OF THE  31 ENTRY POINTS.
  THE TEST EFFECTIVENESS RATIO AT THE ENTRY POINT LEVEL IS   .84

  THE FOLLOWING ELEMENTS WERE NOT CALLED -
   ADBARV  MDBARV  RNIT    SINGLE  SUVW
```

TABLE 4-7. QAPROC Nominal Display

```
        *** PROGRAM CARD IMAGE (KPROG TAPE) LISTING ***

STATEMENT********************          / ELEMENT MAIN   /        PSEUDO NUMBERS  1,   2,   3,   4-FREQ.  1,    2,    3,   4-TYPE

ELEMENT MAIN
      DIMENSION  STATE(8) , ITITLE(14) , X(3),Y(3),Z(3)                           .                      .                   .
      COMMON /MODS/ MOD2,MOD3,MOD5,MOD7                                           .                      .                   .
      DATA  STATE / 8*0. / , I1 / 8 / , I2 / 9 /                                  .                      .                   5
      INPUT = 0                                                                   .                      .                   6
      REWIND I1                                                                   1                      1                  19
      READ (5,100)  ITITLE                                                        2                      1                   0
  100 FORMAT ( 14A6 )                                                             3                      1                  17
  110 READ(5,500) MODEL, (STATE(I),I=1,8)                                         .                      .                   .
              .                                                                   4                      6                  17
              .
              .
      CALL XYZ(X,Y,Z,MODEL)                                                      27                      6                   3
      IF ( INPUT.LT.6 )   GO TO 110                                              28,  29                 6,   5T,   1F      13
      STOP                                                                       30                      1                   4
      END                                                                         .                      .                   .
    .
    .
    .
       ***** PROGRAM STRUCTURE TABLES *****
       7          1
   100300      10016
   101000      20239
    .
    .
    .
      25          3          0        31       1810        1
MAIN        3000030          1
XYZ        24100271         26
 .             .             .
 .             .             .
INTG        1701637       1797
      ***************************************
```

TABLE 4-8a. QAPROC Optional Display (KPROG)

```
          ***** QAPROC INPUT DESCRIPTION *****
 NAME   VALUE     DEFINITION

STATRC   27   STATEMENT EXECUTION RECORDING FILE
MAPSAV    0   OUTPUT MAP SUMMARY FILE
MAPIN.    0   INPUT MAP SUMMARY FILE
BEGNTR    1   NUMBER PAIR AT WHICH TRACE BEGINS(0=NO TRACE)
ENDTR 100000  NUMBER PAIR AT WHICH TRACE ENDS  (0=NO TRACE)
IMAP      0   MAP DISPLAY WILL BE OUTPUT
MAXPAR 100000 NUMBER PAIR AT WHICH TO DISPLAY FIRST MAP
INCPAR 200000 NUMBER OF PAIRS TO PROCESS BETWEEN MAPS
MAPMAX   10   MAXIMUM NUMBER OF MAP DISPLAYS REQUESTED
NOTAB     0   MAP SUMMARIES WILL BE CUMULATIVE
KLIST     0   DO NOT LIST KPROG1 FILE
NCASES    1   NUMBER OF CASES BEING PROCESSED
KTRACK    0   SUBTRK FILE WILL NOT BE OUTPUT
          ************************************

**QAPROC TRACE PRINT**
    ELEMENT MAIN
   1-   3,  16-  17
    ELEMENT RNIT1
   1-   4,
     5-   9(   8 TIMES)
  10-  14,  21-  34
    ELEMENT MAIN
  18-  18,  22-  25,  19-  21
    ELEMENT RNIT2
     .
     .
     .
    ELEMENT XYZ
 240- 241
    ELEMENT MAIN
  28-  28,  30-  30
```

Preceding page blank

TABLE 4-8b. QAPROC Optional Display (TRACE)

KPROG1 file is a copy of the QAMOD file KPROG, with the addition of data relative to the test data execution being evaluated. The optional file, SUBTRK, is explained in Section 5-4.

Due to the large amount of display output available from QAPROC, only a representative portion of the displays is illustrated in Table 4-7.

If the optional feature to display the contents of the KPROG1 file is selected, the display shown in Table 4-8 is output.

4.1.5 QATRAK (TDEM)

QATRAK is the third program of the TDEM Subsystem. The most basic utilization of QATRAK requires the input data file, KPROG1, which is generated by QAPROC. QATRAK accesses the KPROG1 information, determines unexecuted segments of code, and indicates all possibilities for effecting the execution of those segments. A second data file, SUBTRK, is required if the optional features are specified as described in Section 5.5.

All controls of output format, input files, and the type and detail of analysis performed are specified via the NAMELIST $TRAKIN. For the nominal case, only the NAMELIST specification ($TRAKIN and $END) is necessary, since the nominal values assigned in QATRAK are sufficient. The optional features which may be selected are specified between the $TRAKIN and $END cards as explained in Section 5.5.

Following the $END card, a card with *ALL is input as illustrated in Figure 4-3.

To initiate QATRAK processing and indicate the end of control specifications, the *BEGIN card is input, which is always the last card in the QATRAK input data.

The nominal display output of QATRAK consists of an input definitions table and, for each unexecuted segment, a statement of the reason for non-execution, a display of the possible ways to effect the execution of the segment, and a print of all statements which store the value of the branch control variables which determine a transfer to the segment. The only statements that are displayed are those which involve the computation or input of the branch control variables within the element containing the transfer. Optionally, the source code involving computation of the branch

```
        ***** QATRAK INPUT DESCRIPTION *****
 NAME   VALUE    DEFINITION

KRE        5  SYSTEM CARD READ FILE
KPR        6  SYSTEM PRINT FILE
LTRACK     5  TRACKING LEVEL OF DETAIL
MAXTRK   150  NUMBER OF SUBROUTINE CHANGES FOR TRACK-BACK
        ************************************
   .
   .
   .
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
 * * * * * * * * * * * * * * * * * * * * * * * * *  PROCESSING ELEMENT XYZ     * * * * * *
   .
   .
   .
........LOGIC SEGMENT NOT EXECUTED
          PSEUDO NO.      STATEMENT
               210         1000 DO 1010 I=1,3
               211              X(I) = Y(I) * Z(I)
                .               .
                .               .
                .               .
               237              IBODY = 2
               238              GO TO 2050

        ........REFERENCE TO STATEMENT NUMBER  1000 IN XYZ
                PSEUDO NO.  FREQ.       STATEMENT
                 10,   11    3,   0           IF (MODEL.EQ.MOD3) GO TO 1000

               ........COMPUTATION OF VARIABLE MODEL   IN XYZ
                       PSEUDO NO.  FREQ.       STATEMENT
                            0         0              SUBROUTINE XYZ(X,Y,Z,MODEL)

               ........COMPUTATION OF VARIABLE MOD3    IN XYZ
                       PSEUDO NO.  FREQ.       STATEMENT
                            0         0              COMMON /MODS/ MOD2,MOD3,MOD5,MOD7

........   2 LOGIC SEGMENTS WERE NOT EXECUTED IN XYZ

ACCESSING UNIT 29 TO PROVIDE VARIABLE TRACE-BACK THROUGH ELEMENTS EXECUTED

               ........COMPUTATION OF VARIABLE MOD3    IN ELEMENT MAIN
                       PSEUDO NO. , FREQ.      STATEMENT
                          0          0               COMMON /MODS/ MOD2,MOD3,MOD5,MOD7
                          7          6               MOD3 = MODEL - MOD(MODEL,3)

               ........COMPUTATION OF VARIABLE MODEL   IN ELEMENT MAIN
                       PSEUDO NO.  FREQ.       STATEMENT
                          4          6             110 READ(5,500) MODEL,(STATE(I),I=1,8)
                         27          6               CALL XYZ(X,Y,Z,MODEL)
  .
  .
  .
KPREAD INDICATES THAT ALL SUBROUTINES ON THE PROGRAM TAPE HAVE BEEN READ
QATRAK FINISHED
```

TABLE 4-9. QATRAK Nominal Display

control variables in elements executed prior to the element containing the transfer may also be displayed.

Due to the large amount of display output available from QATRAK, only a representative portion of the displays are illustrated in Table 4-9.

If the optional feature to trace external computations of the switch variables is selected, additional information is printed as shown in Table 4-10.

```
        ***** QATRAK INPUT DESCRIPTION *****
 NAME  VALUE    DEFINITION

KRE        5  SYSTEM CARD READ FILE
KPR        6  SYSTEM PRINT FILE
LTRACK     2  TRACKING LEVEL OF DETAIL
MAXTRK   150  NUMBER OF SUBROUTINE CHANGES FOR TRACK-BACK
        ************************************
   .
   .
   .
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
 * * * * * * * * * * * * * * * * * * * * * * * * *  PROCESSING ELEMENT XYZ     * * * * * * * * * * * * * *
   .
   .
   .

              ........COMPUTATION OF VARIABLE MODEL  IN XYZ
                      PSEUDO NO.  FREQ.       STATEMENT
                           0        0                  SUBROUTINE XYZ(X,Y,Z,MODEL)
      *** VARIABLE CORRELATION - CODES = 000001000001 000001000004      NAME = MODEL  ,SUBROUTINE XYZ

              ........COMPUTATION OF VARIABLE MOD3   IN XYZ
                      PSEUDO NO.  FREQ.       STATEMENT
                           0        0                  COMMON /MODS/ MOD2,MOD3,MOD5,MOD7
      *** VARIABLE CORRELATION - CODES = 000000000002 222411300505      NAME = MOD3   ,SUBROUTINE XYZ

........   2 LOGIC SEGMENTS WERE NOT EXECUTED IN XYZ

ACCESSING UNIT 29 TO PROVIDE VARIABLE TRACE-BACK THROUGH ELEMENTS EXECUTED

     CONTROL WAS TRANSFERRED TO XYZ    VIA A RETURN STATEMENT AT PSEUDO NUMBER   18 IN SUBROUTINE STXYZ
     NO VARIABLES ARE BEING TRACKED IN SUBROUTINE STXYZ

     ENTRY  2 OF SUBROUTINE STXYZ  WAS CALLED FROM PSEUDO NUMBER  221 OF XYZ
     .
     .
     .
     ENTRY  1 OF SUBROUTINE XYZ    WAS CALLED FROM PSEUDO NUMBER   27 OF MAIN
              ........COMPUTATION OF VARIABLE MOD3   IN ELEMENT MAIN
                      PSEUDO NO.  FREQ.       STATEMENT
                          0          0                 COMMON /MODS/ MOD2,MOD3,MOD5,MOD7
                          7          6                 MOD3 = MODEL - MOD(MODEL,3)
      *** VARIABLE CORRELATION - CODES = 000000000002 222411300505      NAME = MOD3   ,SUBROUTINE MAIN

              ........COMPUTATION OF VARIABLE MODEL  IN ELEMENT MAIN
                      PSEUDO NO.  FREQ.       STATEMENT
                          4          6             110 READ(5,500) MODEL,(STATE(I),I=1,8)
                         27          6                 CALL XYZ(X,Y,Z,MODEL)
      *** VARIABLE CORRELATION - CODES = 000200010001 000033000004      NAME = MODEL  ,SUBROUTINE MAIN
   .
   .
   .
KPREAD INDICATES THAT ALL SUBROUTINES ON THE PROGRAM TAPE HAVE BEEN READ
QATRAK FINISHED
```

TABLE 4-10. QATRAK Optional Display

5. OPTIONAL FEATURES

The available options for each of the AVS programs are explained in this section. The options have been categorized according to their function: file control, format control, or processing control. For each parameter, a nominal value is included in brackets []. This nominal value will be assigned to the parameter if none is input. If the parameter being defined controls specific output a reference is given to the illustration of the related display.

Due to the complexity of the optional features available for QAMOD, QAPROC, and QATRAK, the "Optional Features Selection Charts" (Figures 5-1 and 5-2) have been provided at the end of this section as an additional aid. Also included is the "AVS File Utilization," Table 5-1, which lists all of the files used by the AVS programs. This table should be referenced if any of the file control parameters are being input.

5.1 TABGEN

5.1.1 <u>File Control</u>

KRE [5] KPR [6]	KRE and KPR are the units accessed by the UNIVAC 1108 card reader and printer. If another computer is being used, KRE and KPR may be changed to correspond to the units used by the new computer.
KSEGBT [3]	=3, output segment relationship data on unit C for input to RAMBLE (Table 4-3) =0, do not generate RAMBLE data
KSFILE [0]	=0, do not generate Program Anatomy Table =8, generate Program Anatomy Table and output the data on Unit F (Table 4-4)

5.2 RAMBLE

The RAMBLE controls are input via a card which is input following the execute card. If the nominal values are desired, a blank card should be input. The options are:

TYPE [0]	Input in card column 5. =0, list paths only, but do not generate the characteristic paths (Table 4-5) =1, generate and display the characteristic paths; list all paths, grouping them with the appropriate characteristic path; display the characteristic loops(Table 4-6)

Input Symbol	File Definition	Unit	QAMOD	Subject Program	TABGEN	RAMBLE	QAPROC	QATRAK
OLDSRC	Subject program source	A	I					I
KSEGBT	"Segments branched to" data	C			O	I		
KSFILE	Program Anatomy Table (optional)	F			O			
NEWSRC	Instrumented subject program source	I	O	P				
KPROG	Subject program structure description	M	O		I		I	
none	AVS programs	N	P		P	P	P	P
KPROG1	Subject program structure description with statement usage frequency data	O					O	I
IDRUM1	FH432 drum units	R			N			N
IDRUM2		S			N			N
none	COMMON structure tables	T						N
IDRUM3	FH432 drum units	U			N			N
IDRUM4		V			N			N
none	Entry point tables	W						N
STATRC	Statement execution recording data	X		O			I	
SUBTRK	Subroutine execution recording data (optional)	Z					O	I

NOTE: Utilization indicated by I=input, O=output, I/O=both, N=internal, P=PCF

Table 5-1. AVS File Utilization

OPTION [0]	Input in card column 10. =0, display all paths =1, display only those paths which end with one of the terminal segments; do not print the loop paths

5.3 QAMOD

5.3.1 File Control

KRE [5] KPR [6] KPN [-3]	KRE, KPR, and KPN are the units accessed by the UNIVAC 1108 card reader, printer, and card punch. If another computer is being used, KRE, KPR, and KPN may be changed to correspond to the units used by the new computer.
OLDSRC [1]	=1, input subject program source code from logical Unit A =i, input the source code from the logical unit designated by i (see footnote)
NEWSRC [11]	=11, output the instrumented source program to logical unit I =i, output the source code to the logical unit designated by i (see footnote)
NFILES [1]	=1, input one file from the source tape OLDSRC =i, where i indicates the number of files to be input from OLDSRC
KPROG [15]	=15, output the subject program structure description data to unit M =i, output the structure data to the logical unit designated by i (see footnote)

5.3.2 Format Control

NOPRNT [0]	=0, display all nominal print, including summary tables and listing of input subject program with pseudo numbers (Table 4-1b) =1, print only the summary tables from the nominal display

---

Footnote: If the instrumented subject program is to be executed in the same run with QAMOD and its execution requires the use of the nominal QAMOD files, that file may be reassigned to an available unit (Table 5-1).

KLIST [0]	=0, print only the nominal display (Table 4-1b) =1, display the KPROG data (Table 4-2)

5.3.3 Processing Control

ISEG [0]	=0, instrument the subject program for TDEM as designated by the parameter "LEVEL" =1, instrument the subject program for ATDG; this option may also be used to obtain TDEM analysis of segments exercised*.
TRPEND [0]	=0, do not instrument the "END" statements; since the "END" statement cannot be executed in the majority of programs, monitoring its execution may produce misleading statistics from TDEM; if, however, the FORTRAN feature of omitting the "RETURN" and "STOP" statements and allowing the "END" statement is used then the "END" statements should be monitored =1, instrument the "END" statements to monitor their execution, since all subroutines do not transfer control via a "RETURN" or "STOP"
IFA [1]	=1, consider each arithmetic "IF" statement to be four executable statements; instrument such that the execution of the decision and each of the three transfer options be monitored individually =0, instrument each arithmetic "IF" such that only the execution of the decision is monitored without indication of the usage of each transfer NOTE: IFA=1 results in the insertion of 4 statements for instrumentation, whereas IFA=0 requires only one additional statement
NINOPT [1]	=1, input the subject program source code from the tape (unit OLDSRC) =0, read the source code from cards which are input following the *BEGIN card; each element must be preceded by an identification card with "ELEMENT" in Columns 1-7 and the element name in Columns 9-14

* This technique results in assigning the same importance to all segments regardless of their size. That is, if ISEG = 0, a segment of code consisting of 20 statements would be 10% of a 200 statement program. If ISEG = 1 and the program contained 4 segments, the 20 statement segment would be 25% of the program and be considered equally as important as a segment of 100 statements.

NOTOPT [1]	=1, output the instrumented source code to tape (unit NEWSRC) =0, punch the instrumented source on cards NOTE: The card output should be used with caution since the instrumented source is substantially larger than the input source program
LEVEL [1]	=1, instrument such that the execution of all statements in the subject program will be monitored ≠1, instrument according to the indicated level of detail; available levels are explained in Table 5-2. NOTE: QATRAK is only operational on a program processed with LEVEL = 1.
LIMSEG [0]	=0, instrument as indicated by "LEVEL" =1, instrument the statements ordinarily processed if LEVEL=3, plus all labelled statements which follow an unconditional transfer (arithmetic IF, GO TO, RETURN, STOP, or CALL CHAIN)
Element Specification Cards	These cards are input instead of the *ALL card following the $END card. Each element being specified is input on a separate card, with the name beginning in card column 1. A maximum of 20 cards may be input. If LEVEL equals 1, 2, or 3, only the elements specified will be processed. If LEVEL equals 11, 12, or 13, all elements except those specified will be processed. Other levels are explained in Table 5-2.

5.4 QAPROC

5.4.1 File Control

STATRC [27]	=27, input the statement execution recording data from unit X =i, input the data from the logical unit indicated by i; this input is required when more than one recording tape is being processed by a single QAPROC execution; available units may be derived from Table 5-1.
MAPSAV [0] MAPIN [0]	See explanations under "Processing Control"

TABLE 5-2. QAMOD Instrumentation Levels

Analysis Level	Statements Processed In Selected Elements*	Statements Processed In Unselected Elements*
1	All executable	None (element skipped)
2	(1) First executable in element (2) STOP (or CALL EXIT) (3) CALL CHAIN (...) (4) First executable after each entry point (5) END (6) RETURN (7) Labelled statement (except loop terminator) (8) CALL (if referenced element is analyzed) (9) GO TO (10) Arithmetic IF	None
3	Numbers 1-6 of level 2	None
4	All executable	Numbers 1-4 of level 2
5	All executable	Numbers 1-3 of level 2
6	All executable**	Numbers 1&4 of level 2
7	All executable**	Number 1 of level 2
8	Numbers 1-4 of level 2**	Numbers 1&4 of level 2
9	Numbers 1-3 of level 2**	Number 1 of level 2

* If the analysis level is input as defined above ($1 \le \text{LEVEL} \le 9$), selected elements are those specified by the QAMOD element specification cards.

If the indicated level plus ten is input ($11 \le \text{LEVEL} \le 19$), selected elements are those not specified by element specification cards (e.g., if LEVEL = 11, all elements will be processed at level 1 except those specified on the cards).

** At least one element must be specified at this level.

28

5.4.2 Format Control

BEGNTR [0]	=0, no "TRACE" output is desired (Table 4-8b) =i, begin display of "TRACE" print when the number of pairs of statement execution recordings processed equals i (the total number of recordings output is printed on the last page of the execution of the instrumented subject program)
ENDTR [0]	=0, no "TRACE" output is desired =i, end display of "TRACE" print when the number of pairs of recordings processed equals i
IMAP [0]	=0, display nominal "MAP" summary (Table 4-7) =1, do not display "MAP" summary
MAXPAR [100000]	=100000, display the initial "MAP" when 100,000 pairs of statement execution records has been processed (Table 4-7) =i, display the initial "MAP" when the number of recording pairs processed equals i.
INCPAR [200000]	=200000, display intermediate "MAP" summaries after each 200,000 recording pairs processed (nominal values generate displays at 100000, 300000, 500000, etc.) =i, display intermediate "MAP" summaries at intervals of i recording pairs
MAPMAX [10]	=10, output a maximum of 10 "MAP" summary displays =i, output a maximum of i "MAP" summary displays
NOTAB [0]	=0, output cumulative execution frequencies when processing multiple recording files =1, reinitialize frequencies at zero before processing each recording file
KLIST [0]	=0, do not list the KPROG1 file =1, list the KPROG1 file (Table 4-8a)

5.4.3 Processing Control

NCASES [1]	=1, process one statement execution recording file from tape "STATRC" =i, process i recording files
KTRACK [0]	=0, do not generate a subroutine execution recording file for QATRAK (the parameter LTRACK must then be set equal to zero or one during QATRAK execution)

KTRACK [0] (Continued)

=1, generate a SUBTRK file on Unit 29 (Z) for QATRAK (execution of QATRAK is then performed with LTRACK ≥ 2)

MAPSAV [0]

=0, do not save the "MAP" summary data on an output file
=i, output the "MAP" data to the unit indicated by i; "MAP" data is saved to be accumulated with additional recording data at a subsequent QAPROC execution

MAPIN [0]

=0, initialize the internal "MAP" array with zeros
=i, read the "MAP" summary data from unit i and store the data in the internal "MAP" array before processing the STATRC data; this option is used with "MAPSAV" to accumulate frequencies from several executions

5.5 QATRAK

5.5.1 File Control

KRE [5]
KPR [6]

KRE and KPR are the units accessed by the UNIVAC 1108 card reader and printer. If some other computer is being used, the unit may be input to correspond to the new computer.

5.5.2 Format Control

LTRACK

=0, display unexecuted segments and decision variables involved, but do not track the computation and input of the variables.
=1, track the computation and input of decision variables within the subroutine being processed.
=2, print all of the level 1 information, then access the subroutine execution trace data from file 29 (unit Z) and display:

a) the calculation and input of the decision variables in previously executed subroutines,
b) variable correlation information for each subroutine in which variables are referenced,
c) walk-back information giving the pseudo number from which control was transferred in each subroutine executed prior to the original element,

d) explanatory information for each subroutine in which no variables are tracked.

=3, all of level 2 except d).
=4, all of level 2 except c) and d).
=5, all of level 2 except b), c), and d).
NOTE: Additional explanation of LTRACK are provided in Section 5.5.4.
LEVEL = 5 is illustrated in Table 4-9 and
LEVEL = 2 in Table 4-10.

5.5.3 Processing Control

MAXTRK [150]	If LTRACK ≥ 2, the computation of decision variables is tracked through each subroutine executed prior to the execution of the decision being described. The parameter MAXTRK is input to terminate the trace-back after a particular number of subroutines have been checked since the same subroutine may be entered several times causing redundant checking. To stop the trace-back after 'n' subroutine transfers, input MAXTRK = n. To allow a traceback through the entire execution of the subject program, input MAXTRK = -1. The nominal value of 150 should be sufficient for the majority of programs. Note: the trace-back may be further controlled for specific elements by use of the element specification card explained below.
Element Specification Card	If particular elements (maximum 20) require special processing, they may be specified on cards after the $END terminator of $TRAKIN. A description of the fields of these cards is provided below.

Card Column	Variable	Description
1-6	NAME	Alphanumeric name of the element being specified (left-justified).
7-11	STRTRK	When variable computations are tracked through more than one subroutine (LTRACK ≥ 2), the trace-back is initiated at the last call to the subroutine being processed. If STRTRK is set to n, then trace-back is initiated 'n' calls prior to the termination of the subject program execution.
7-12	MAXTRK	Same definition as the $TRAKIN parameter MAXTRK. The value input is functional only during the processing of element 'NAME'.

5.5.4 Additional Explanations of LTRACK Display Levels

- LTRACK = 1,2,3,4,5

  Each subroutine is scanned and for each unexecuted logic segment the following is printed:

  > The statements in the unexecuted segment with their pseudo numbers;
  >
  > Each statement, within the subroutine, from which the unexecuted segment could have been accessed;
  >
  > The control variables responsible for the decision not to branch to the segment;
  >
  > The statements, within the subroutine, which involved the computation or input of these variables.

  If the segment is inaccessible, information is printed explaining why it is inaccessible (Table 4-9).

- LTRACK = 2,3,4,5

  Elements executed prior to the subroutine being processed are scanned and the following is also print:

  > The statement "ACCESSING UNIT 29 TO PROVIDE TRACE-BACK THROUGH ELEMENTS EXECUTED PRIOR TO THE LAST -j EXECUTION OF ELEMENT XYZ," where j is one less than the value input for STRTRK on the subroutine specification card for subroutine XYZ. This indicates that the subroutine execution trace data file is being accessed to find the final execution of the element XYZ. The file is then scanned backward until the n^{th} execution of XYZ is encountered, where n = STRTRK. The trace-back will then commence at that point in the data file. If STRTRK is not input, it is set to zero.
  >
  > Statements in the subject program which could have caused computation or input of the decision variables being traced. The statement "TRACE-BACK TERMINATED BY PROCESSING OF MAIN PROGRAM YYY," if complete trace is specified ($TRAKIN parameter MAXTRK = -1), or the statement "TRACE-BACK TERMINATED AFTER PROCESSING i SUBROUTINE CHANGES (MAXTRK = i)" if partial trace is specified ($TRAKIN parameter MAXTRK = i).

- LTRACK = 2,3,4

  Display also includes:

  > Variable correlation information for each subroutine in which the variables are referenced (Since the alphanumeric

names of variables may change between subroutines, they have been assigned octal codes as a means of correlating variable names between subroutines. These are printed out as "Correlation Codes," with the current variable and subroutines names).

Example: If the variable X is tracked in SUB1 and then is called Y in SUB2, the following correlations may be printed:

```
***VARIABLE CORRELATION-CODES = 262145 327444, NAME=X, SUBROUTINE SUB1

***VARIABLE CORRELATION-CODES = 262145 372444, NAME=Y, SUBROUTINE SUB2
```

- LTRACK = 2,3

  Information is displayed each time a subroutine is checked, whether additional tracking data is generated or not. This additional information includes:

  Listing of statements which could have stored the values of the variables being tracked, or

  The statement "VARIABLE XXX HAS BEEN PREVIOUSLY TRACKED IN ELEMENT ABC," or

  The statement "NO VARIABLES ARE BEING TRACKED IN SUB-ROUTINE ABC," if subroutine ABC does not contain any of the global COMMON or calling sequence variables being tracked.

- LTRACK = 2

  Walk-back information is displayed, giving the pseudo number from which control was transferred in each element executed prior to the original element (Table 4-10).

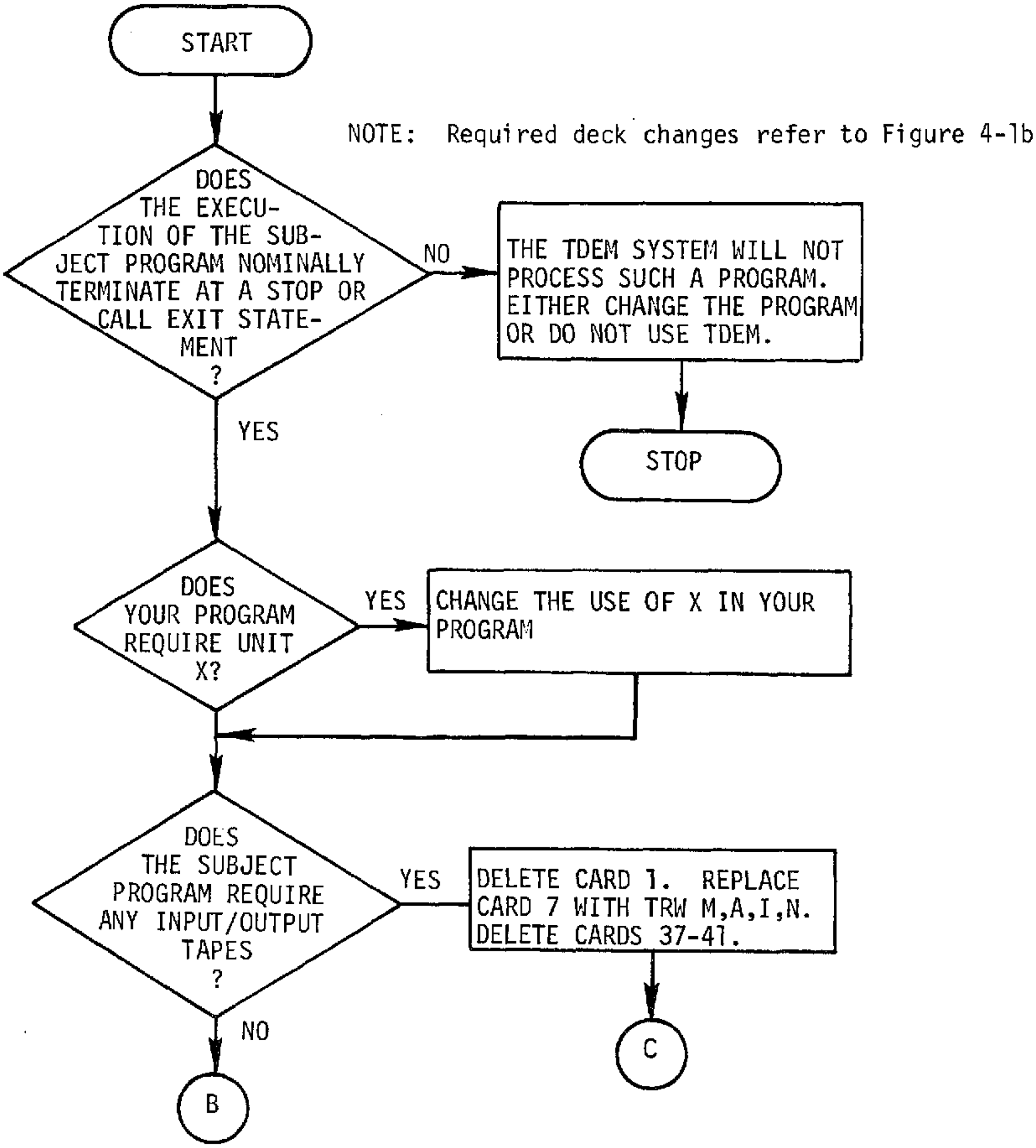

Figure 5-1. QAMOD (TDEM) Optional Features Selection Chart

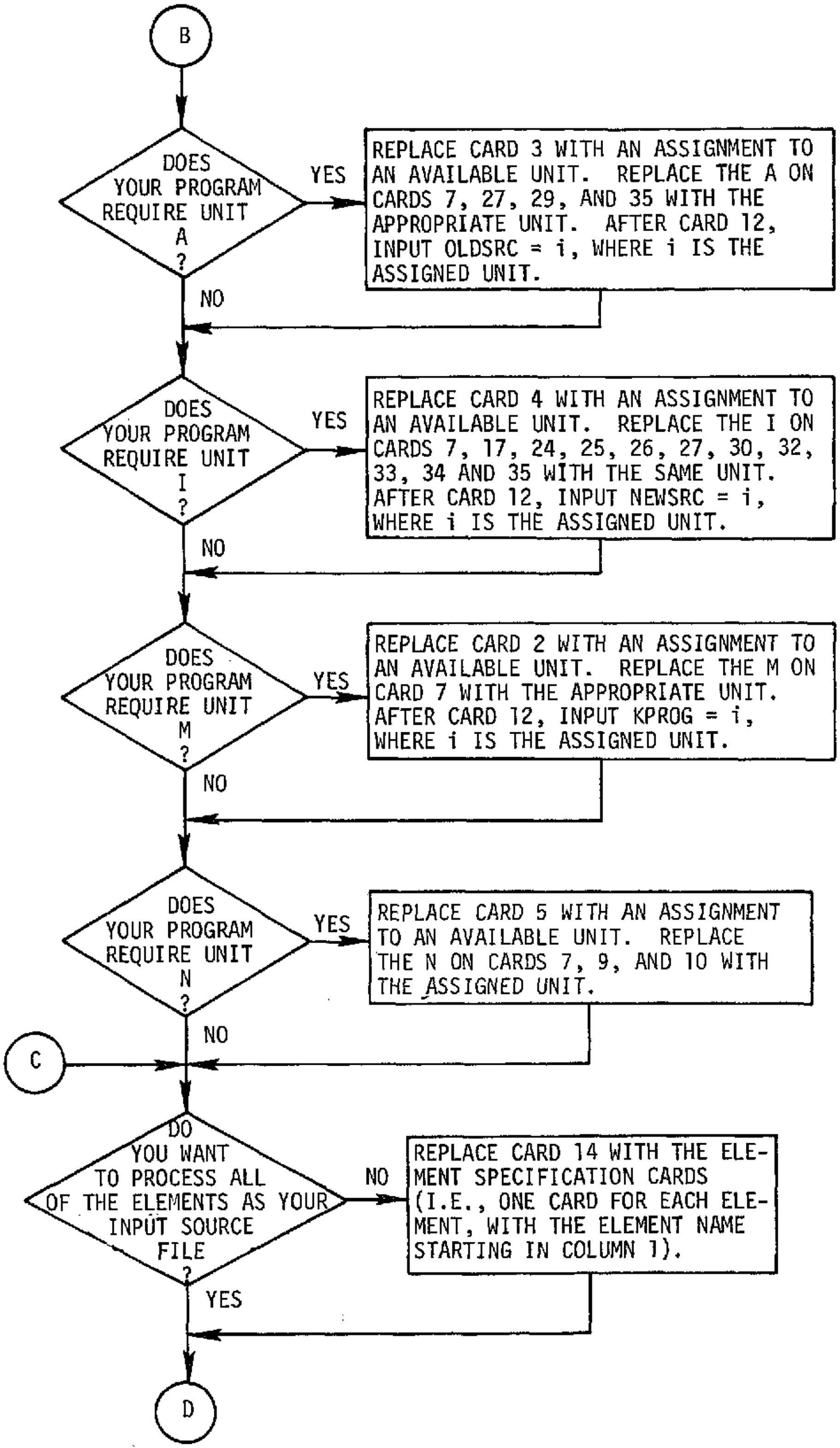

Figure 5-1. QAMOD (TDEM) Optional Features Selection Chart (Continued)

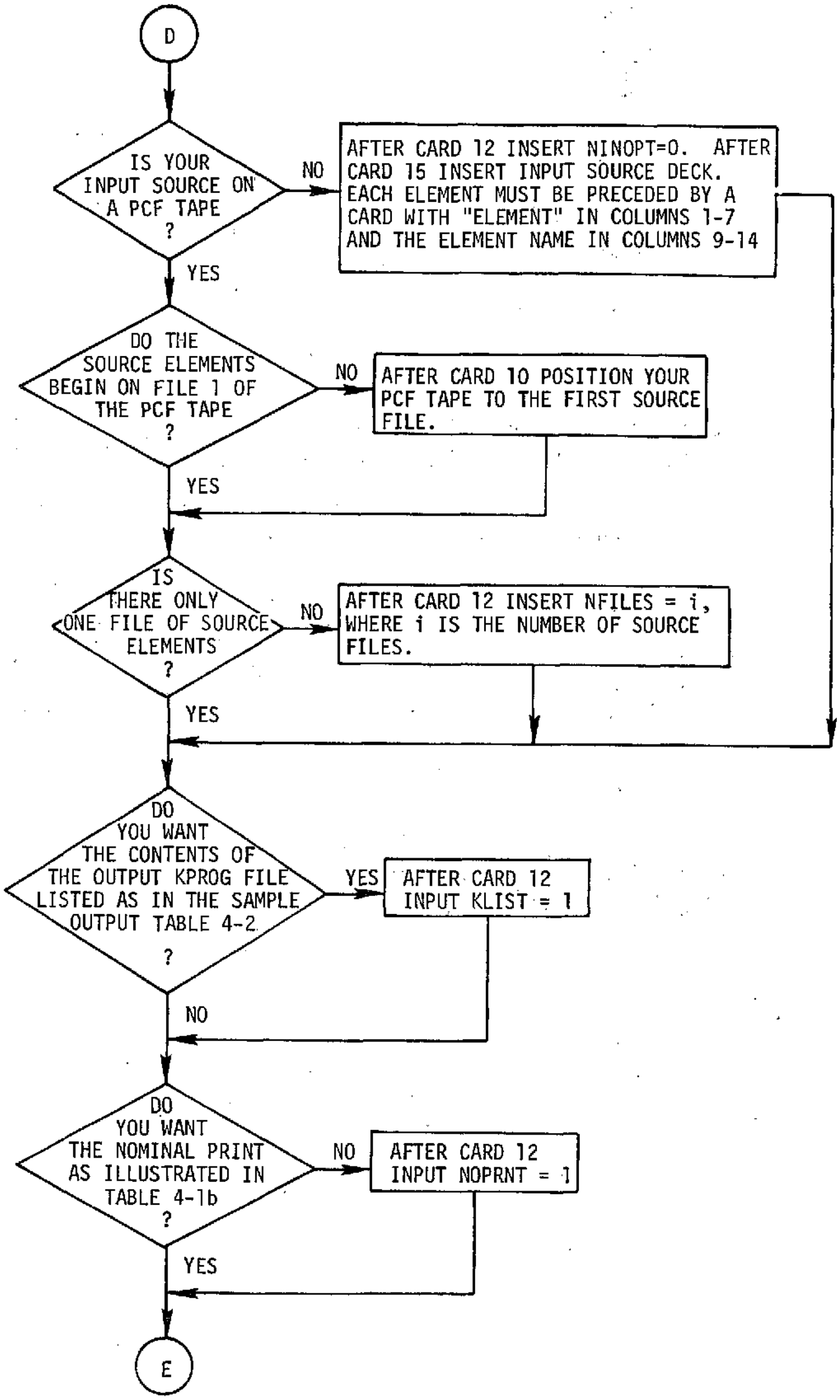

Figure 5-1. QAMOD (TDEM) Optional Features Selection Chart (Continued)

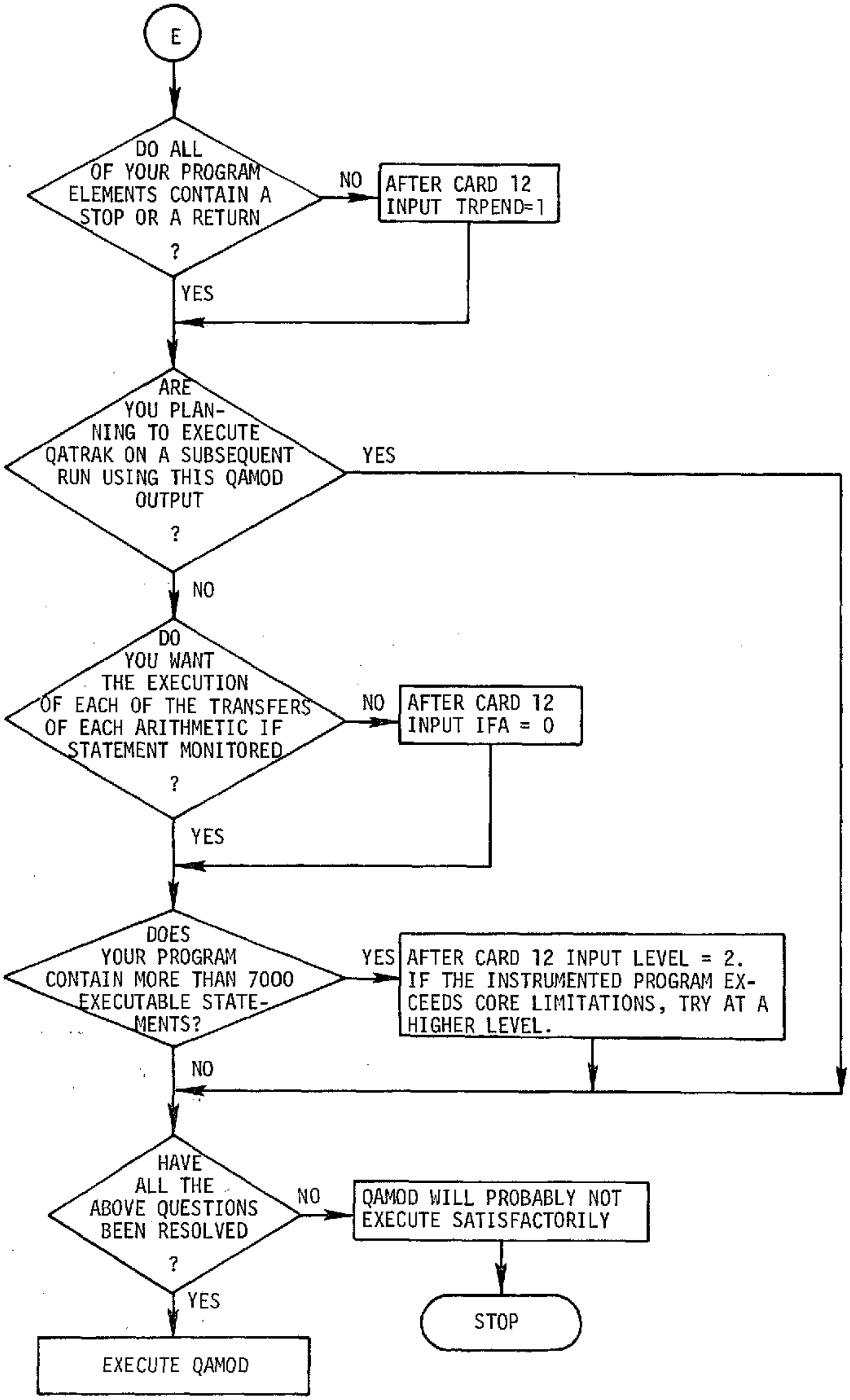

Figure 5-1. QAMOD (TDEM) Optional Features Selection Chart (Continued)

NOTE: Required deck changes refer to Figure 4-3.

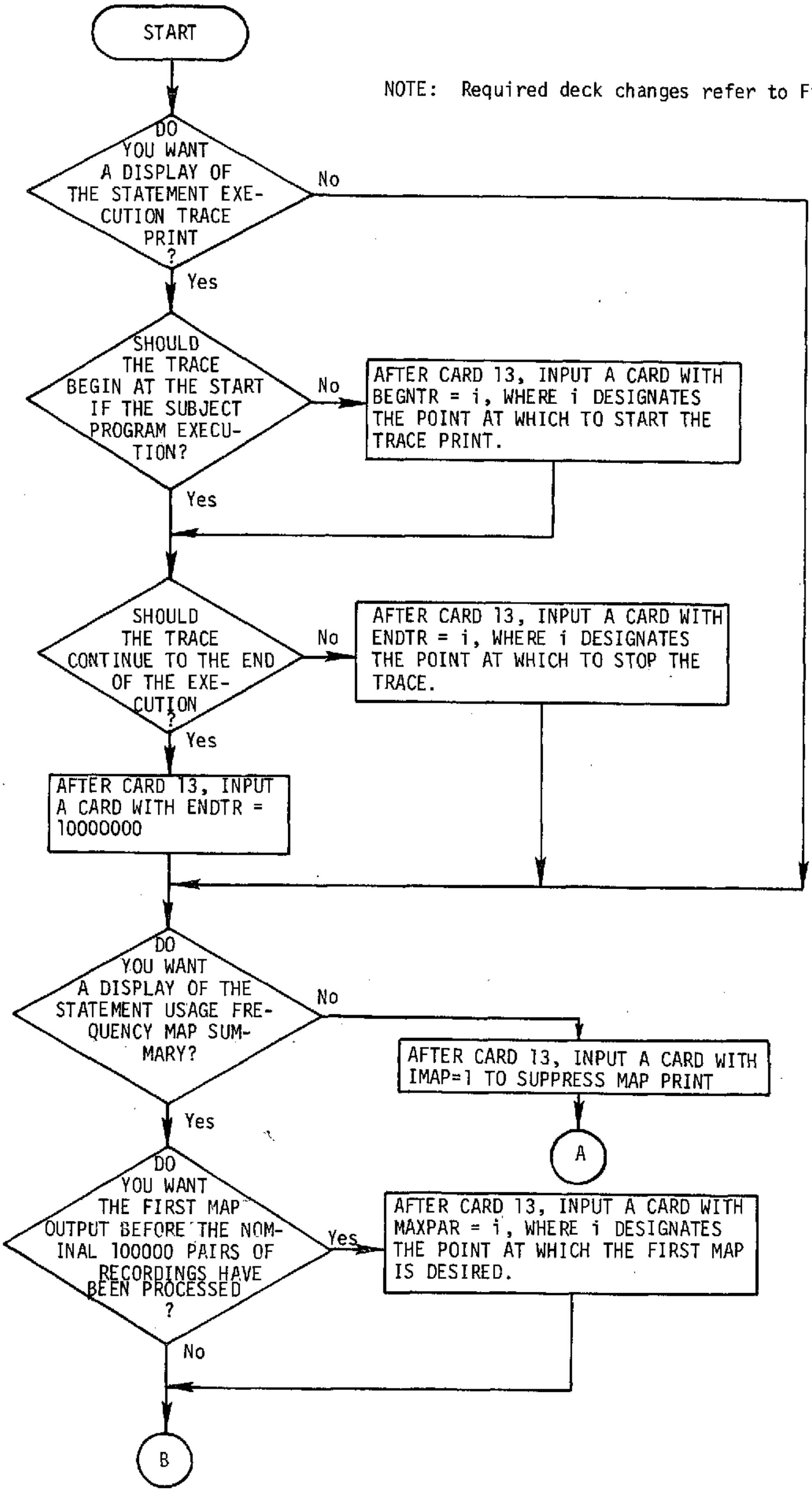

Figure 5-2. QAPROC/QATRAK Optional Features

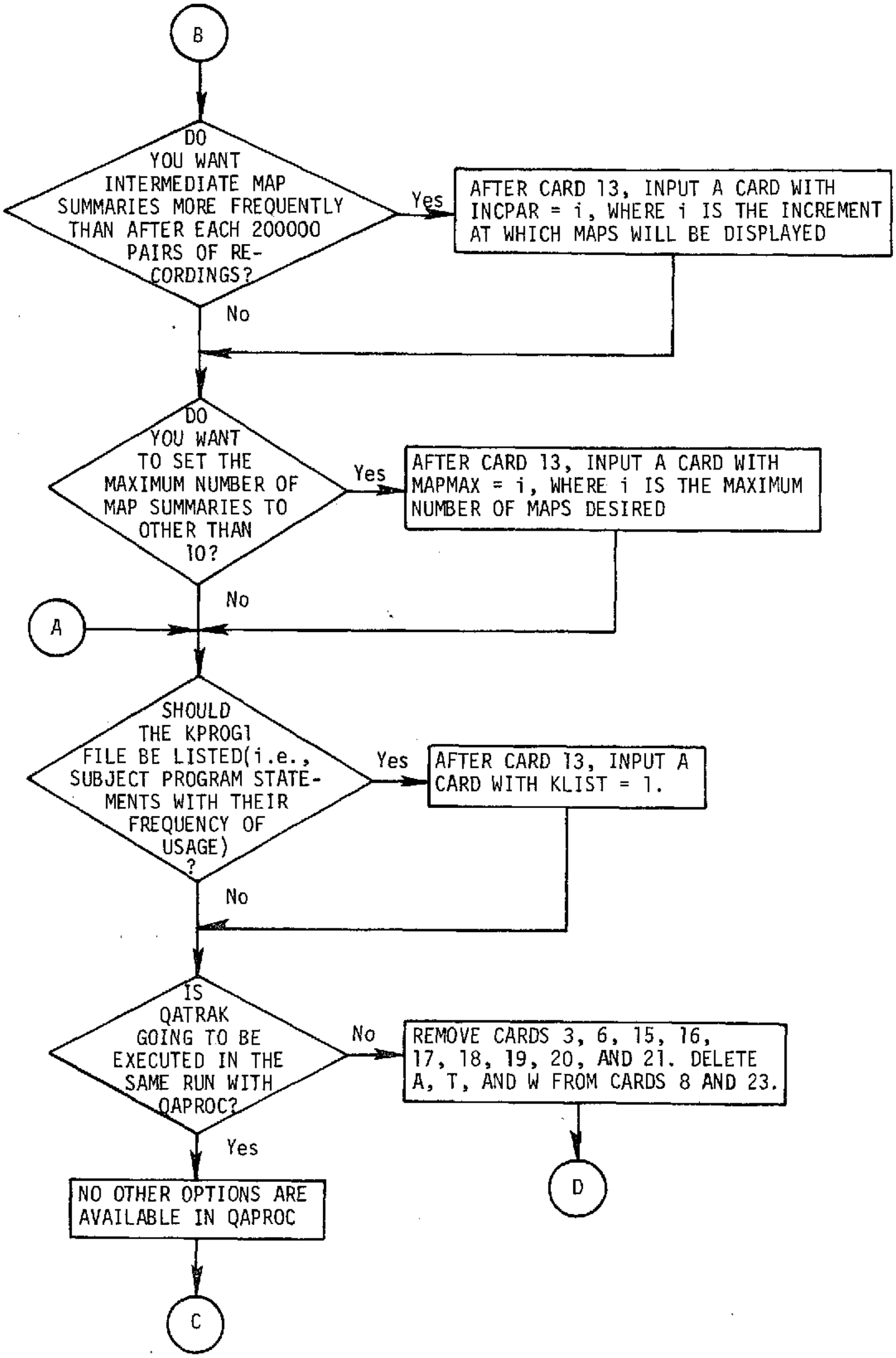

Figure 5-2. QAPROC/QATRAK Optional Features (Continued)

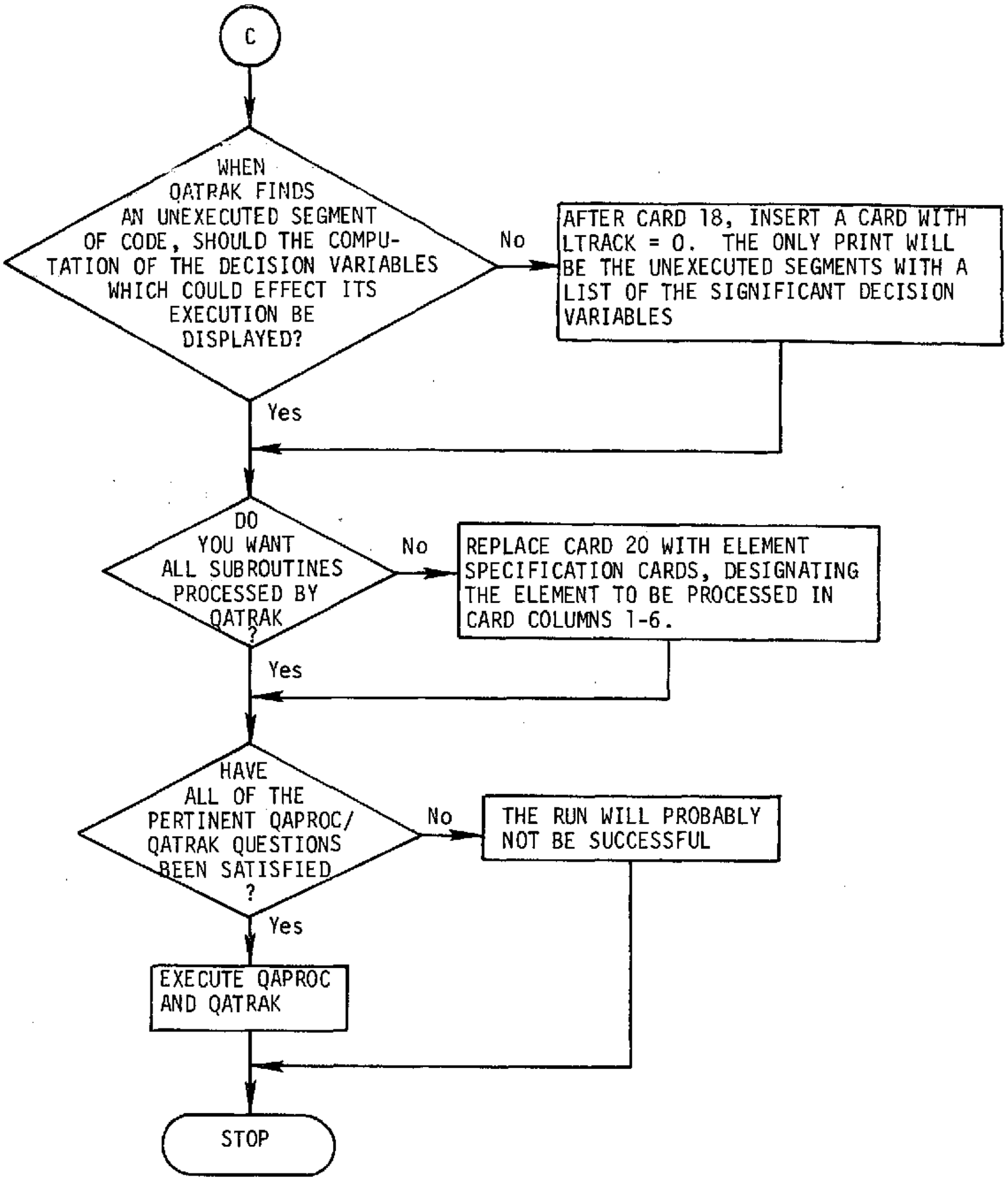

Figure 5-2. QAPROC/QATRAK Optional Features (Continued)

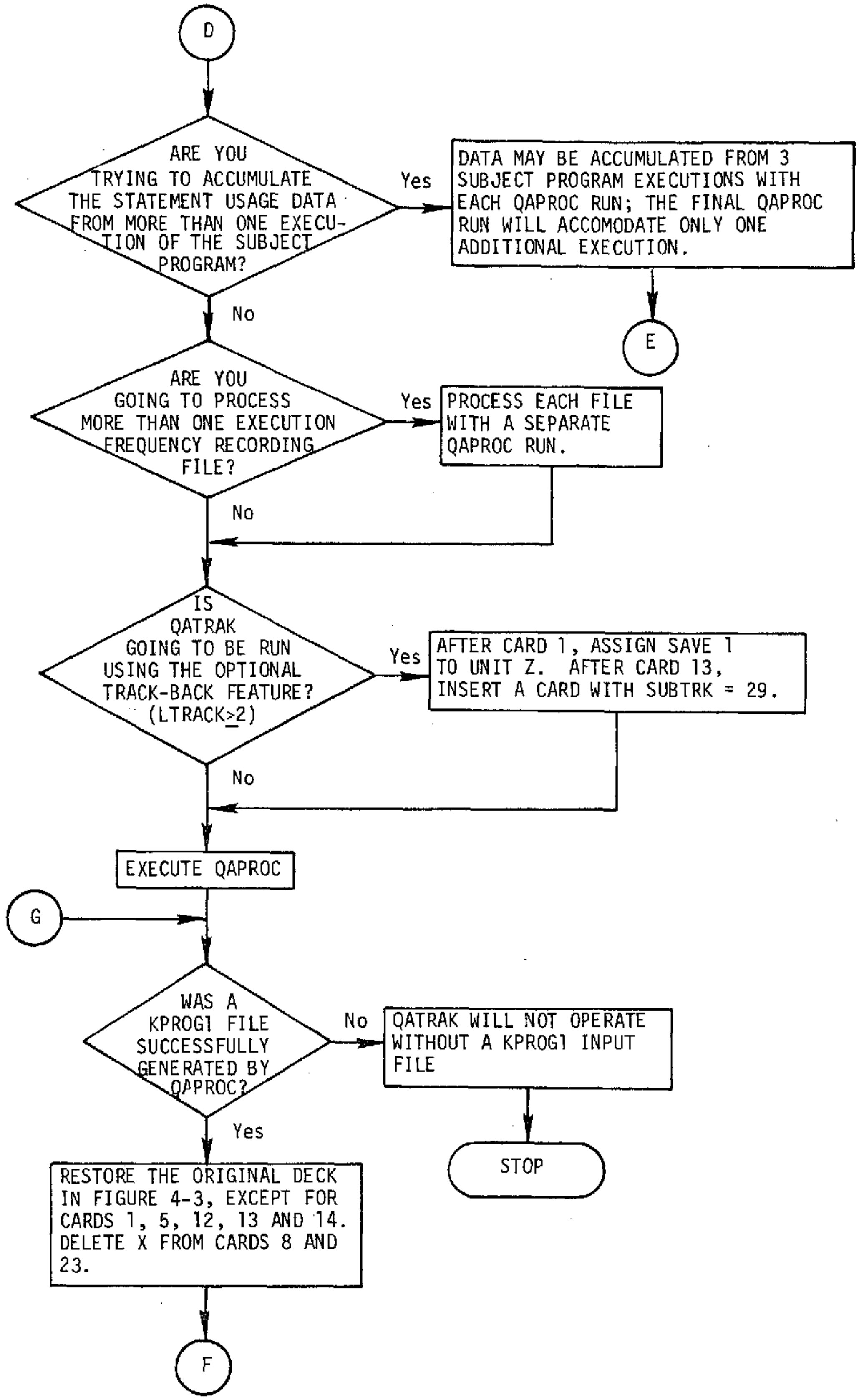

Figure 5-2. QAPROC/QATRAK Optional Features (Continued)

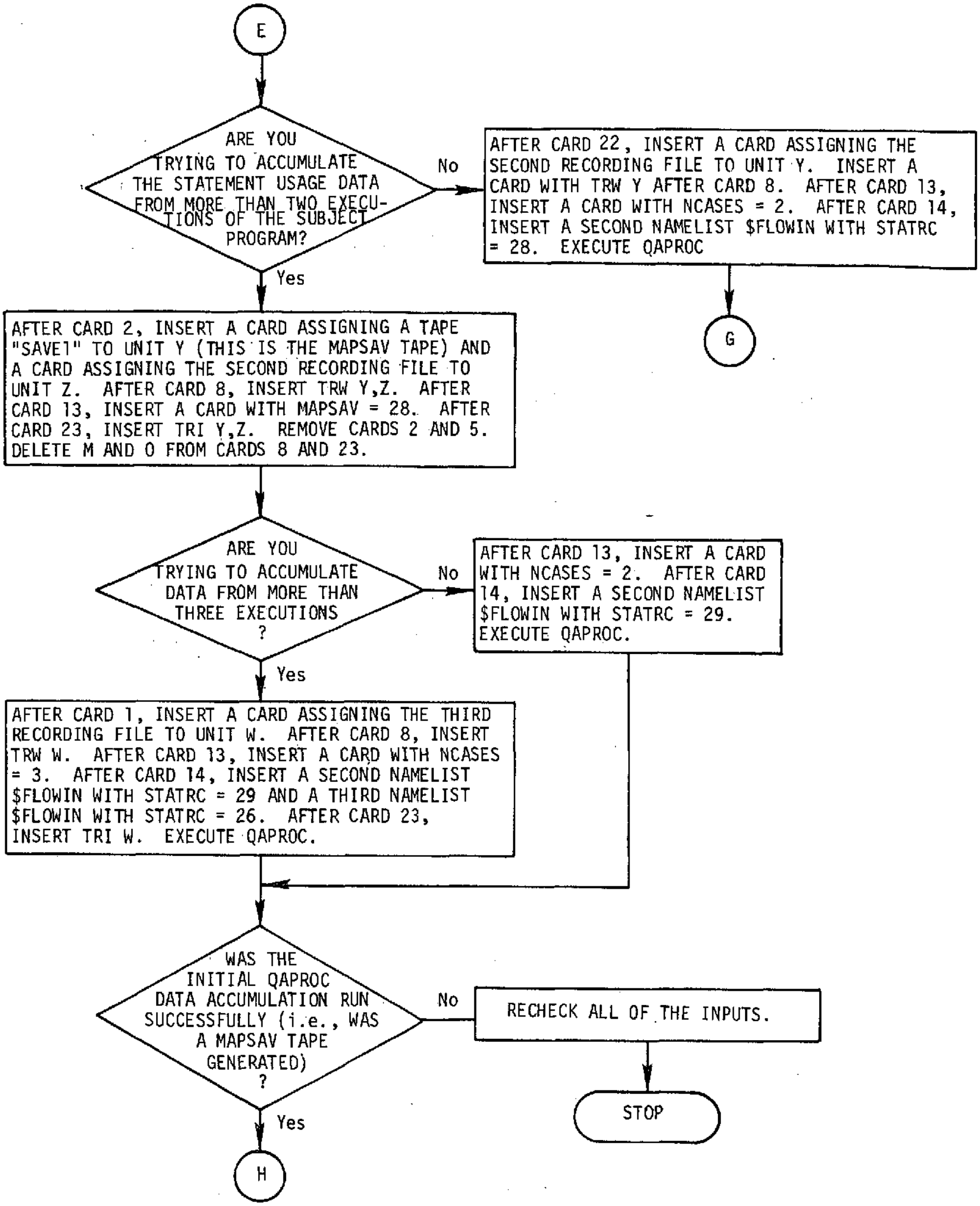

Figure 5-2. QAPROC/QATRAK Optional Features (Continued)

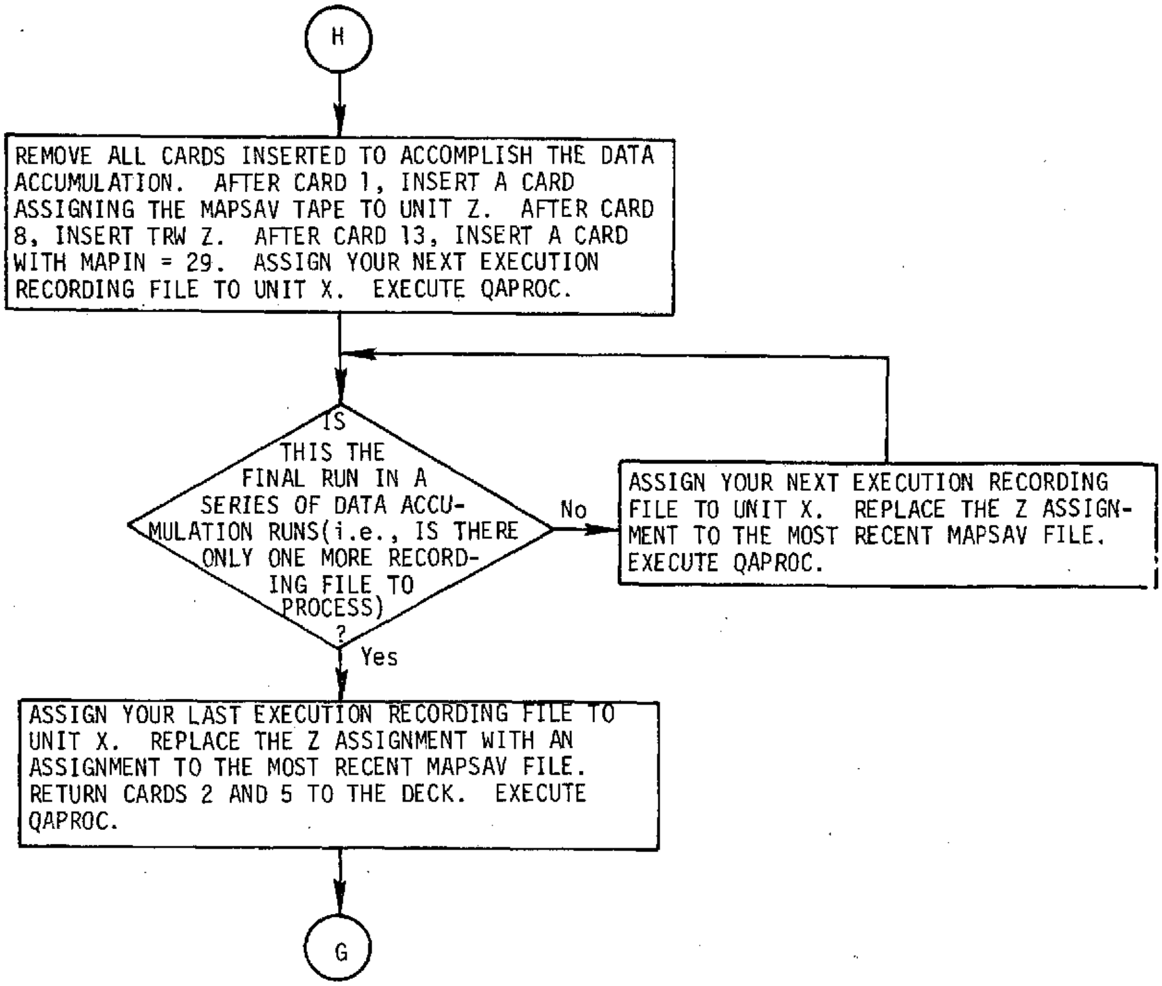

Figure 5-2. QAPROC/QATRAK Optional Features (Continued)

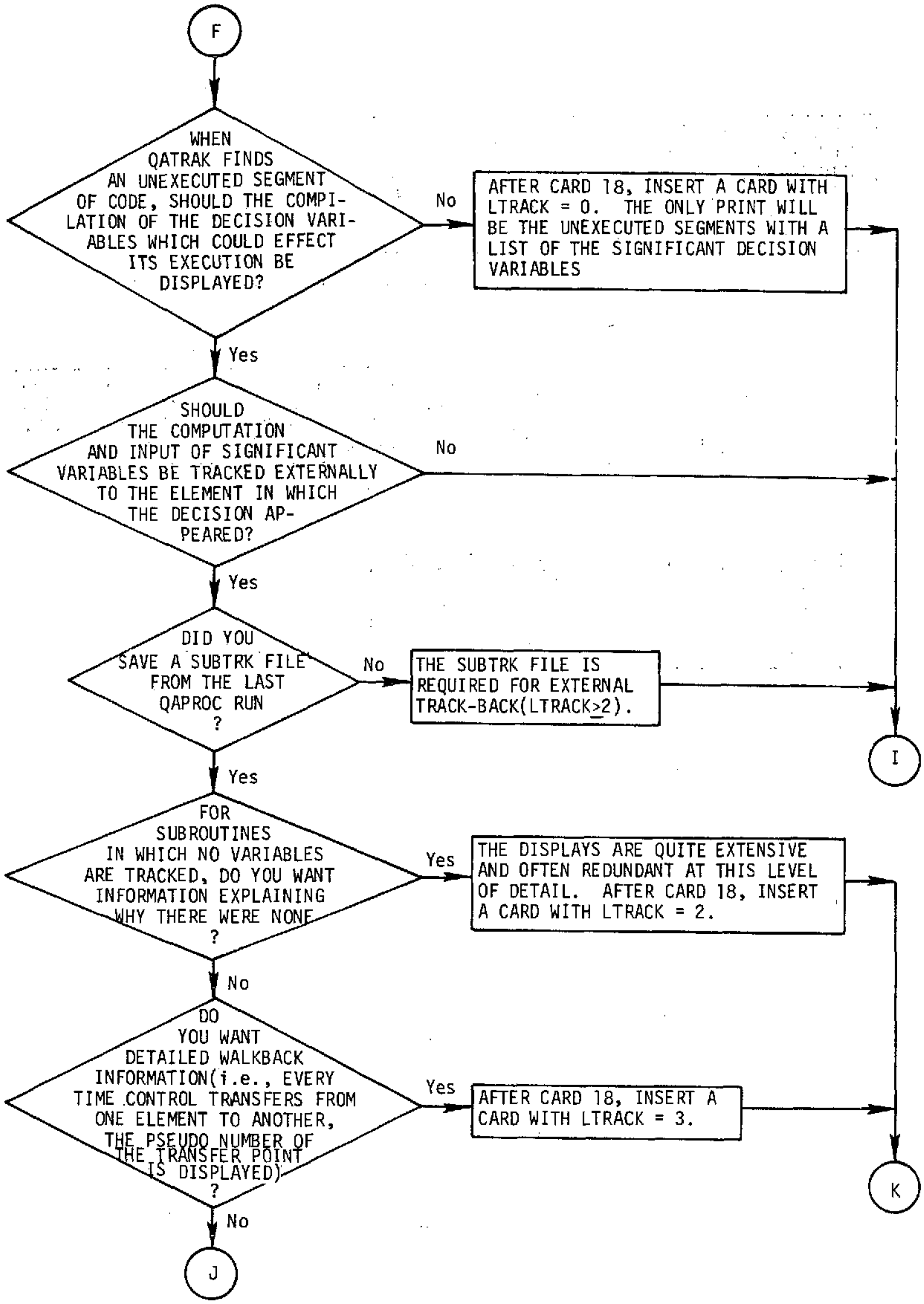

Figure 5-2. QAPROC/QATRAK Optional Features (Continued)

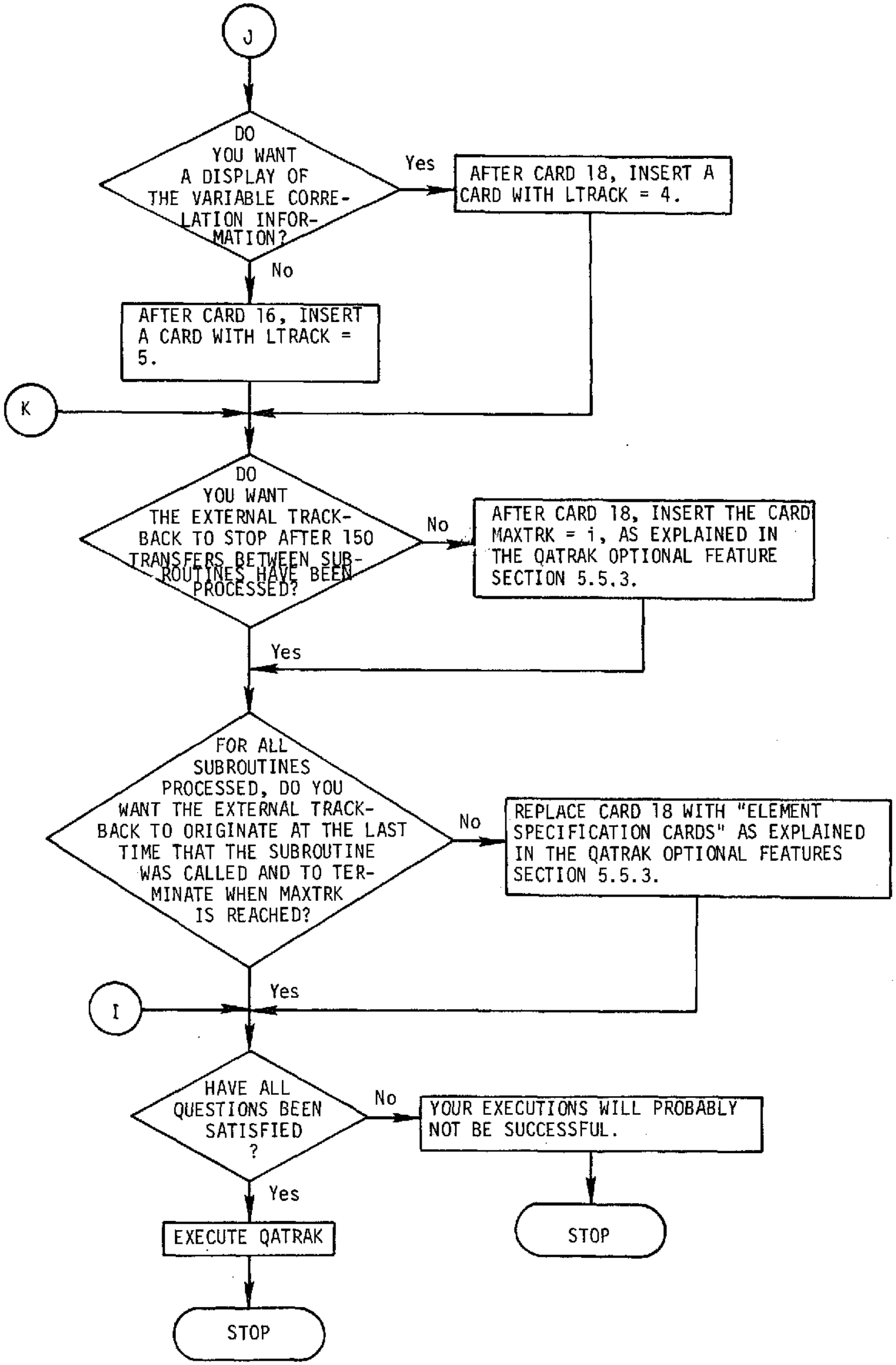

Figure 5-2. QAPROC/QATRAK Optional Features (Continued)

REFERENCES

1. C. G. Gibson and L. R. Railing, "Verification Guidelines," TRW 70-FMT-884, (17618-H200-RO-00), 27 August 1971.

2. K. W. Krause, "User's Guide for Updated Version of the Test Data Effectiveness Measurement System (TDEM)," TRW IOC 71-4912.30-57, 1 July 1971.